TOSHIBA

64-Bit TX System RISC
TX49 Family
TMPR4938

Rev. 2.0

TOSHIBA CORPORATION

R4000/R4400/R5000 are a trademark of MIPS Technologies, Inc.

The information contained herein is subject to change without notice.

The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others.

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress.
It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..

The Toshiba products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).
These Toshiba products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of Toshiba products listed in this document shall be made at the customer's own risk.

The products described in this document may include products subject to the foreign exchange and foreign trade laws.

The products described in this document contain components made in the United States and subject to export control of the U.S. authorities. Diversion contrary to the U.S. law is prohibited.

TOSHIBA products should not be embedded to the downstream products which are prohibited to be produced and sold, under any law and regulations.

© 2005 TOSHIBA CORPORATION
All Rights Reserved

Preface

Thank you for new or continued patronage of TOSHIBA semiconductor products. This is the 2005 edition of the user's manual for the TMPR4938 64-bit RISC microprocessor.

This databook is written so as to be accessible to engineers who may be designing a TOSHIBA microprocessor into their products for the first time. No prior knowledge of this device is assumed. What we offer here is basic information about the microprocessor, a discussion of the application fields in which the microprocessor is utilized, and an overview of design methods. On the other hand, the more experienced designer will find complete technical specifications for this product.

Toshiba continually updates its technical information. Your comments and suggestions concerning this and other Toshiba documents are sincerely appreciated and may be utilized in subsequent editions. For updating of the data in this manual, or for additional information about the product appearing in it, please contact your nearest Toshiba office or authorized Toshiba dealer.

March 2005

Table of Contents

Handling Precautions

TMPR4938

1. Overview and Features 1-1
1.1 Overview 1-1
1.2 Features 1-1
1.2.1 Features of the TX49/H3 core 1-2
1.2.2 Features of TX4938 peripheral functions 1-2
2. Configuration 2-1
2.1 TX4938 block diagram 2-1
3. Signals 3-1
3.1 Pin Signal Description 3-1
3.1.1 Signals Common to SDRAM and External Bus Interfaces 3-1
3.1.2 SDRAM Interface Signals 3-2
3.1.3 External Interface Signals 3-3
3.1.4 DMA Interface Signals 3-4
3.1.5 PCI Interface Signals 3-4
3.1.6 Serial I/O Interface Signals 3-6
3.1.7 Timer Interface Signals 3-6
3.1.8 Parallel I/O Interface Signals 3-6
3.1.9 AC-link Interface Signals 3-7
3.1.10 Interrupt Signals 3-7
3.1.11 SPI Interface Signals 3-7
3.1.12 ISA/ATA Interface Signals 3-8
3.1.13 ETHER Interface Signals (Channel 0) 3-8
3.1.14 ETHER Interface Signals (Channel 1) 3-9
3.1.15 NAND Flash Memory Interface Signals 3-10
3.1.16 Extended EJTAG Interface Signals 3-10
3.1.17 Clock Signals 3-11
3.1.18 Initialization Signal 3-11
3.1.19 Test Signals 3-11
3.1.20 Power Supply Pins 3-12
3.2 Boot Configuration 3-13
3.3 Pin multiplex 3-17
4. Address Mapping 4-1
4.1 TX4938 Physical Address Map 4-1
4.2 Register Map 4-2
4.2.1 Addressing 4-2
4.2.2 Ways to Access to Internal Registers 4-2
4.2.3 Register Map 4-3
5. Configuration Registers 5-1
5.1 Detailed Description 5-1
5.1.1 Detecting G-Bus Timeout 5-1
5.2 Registers 5-2
5.2.1 Chip Configuration Register (CCFG) 0xE000 5-3
5.2.2 Chip Revision ID Register (REVID) 0xE008 5-6
5.2.3 Pin Configuration Register (PCFG) 0xE010 5-7
5.2.4 Timeout Error Access Address Register (TOEA) 0xE018 5-10
5.2.5 Clock Control Register (CLKCTR) 0xE020 5-11
5.2.6 G-Bus Arbiter Control Register (GARBC) 0xE030 5-14
5.2.7 Register Address Mapping Register (RAMP) 0xE048 5-15

5.2.8 Jump Address Register (JMPADR) 0xE058 5-16
6. Clocks 6-1
6.1 TX4938 Clock Signals 6-1
6.2 Power-Down Mode 6-5
6.2.1 Halt Mode and Doze Mode 6-5
6.2.2 Power Reduction for Peripheral Modules 6-5
6.3 Power-On Sequence 6-6
7. External Bus Controller 7-1
7.1 Features 7-1
7.2 Block Diagram 7-2
7.3 Detailed Explanation 7-3
7.3.1 External Bus Control Register 7-3
7.3.2 Global/Boot-up Options 7-4
7.3.3 Address Mapping 7-5
7.3.4 External Address Output 7-6
7.3.5 Data Bus Size 7-7
7.3.6 Access Mode 7-9
7.3.7 Access Timing 7-13
7.3.8 Clock Options 7-19
7.3.9 ISA /ATA Mode 7-20
7.4 Register 7-24
7.4.1 External Bus Channel Control Register (EBCCRn) 0x9000 (ch. 0), 0x9008 (ch. 1) 0x9010 (ch. 2), 0x9018 (ch. 3) 0x9020 (ch. 4), 0x9028 (ch. 5) 0x9030 (ch. 6), 0x9038 (ch. 7) 7-25
7.5 Timing Diagrams 7-28
7.5.1 ACE* Signal 7-29
7.5.2 Normal mode access (Single, 32-bit Bus) 7-31
7.5.3 Normal mode access (Burst, 32-bit Bus) 7-35
7.5.4 Normal Mode Access (Single, 16-bit bus) 7-37
7.5.5 Normal Mode Access (Burst, 16-bit Bus) 7-41
7.5.6 Normal Mode Access (Single, 8-bit Bus) 7-43
7.5.7 Normal Mode Access (Burst, 8-bit Bus) 7-46
7.5.8 Page Mode Access (Burst, 32-bit Bus) 7-48
7.5.9 External ACK Mode Access (32-bit Bus) 7-50
7.5.10 READY Mode Access (32-bit Bus) 7-56
7.5.11 ISA IO Space Access 7-58
7.5.12 ATA/PIO Transfer Mode Access 7-60
7.6 Flash ROM, SRAM Usage Example 7-62
8. DMA Controller 8-1
8.1 Features 8-1
8.2 Block Diagram 8-2
8.3 Detailed Explanation 8-4
8.3.1 Transfer Mode 8-4
8.3.2 On-chip Registers 8-5
8.3.3 External I/O DMA Transfer Mode 8-5
8.3.4 Internal I/O DMA Transfer Mode 8-8
8.3.5 Memory-Memory Copy Mode 8-9
8.3.6 Memory Fill Transfer Mode 8-9
8.3.7 Single Address Transfer 8-9
8.3.8 Dual Address Transfer 8-12
8.3.9 DMA Transfer 8-17
8.3.10 Chain DMA Transfer 8-18
8.3.11 Dynamic Chain Operation 8-20
8.3.12 Interrupts 8-21
8.3.13 Transfer Stall Detection Function 8-21
8.3.14 Arbitration Among DMA Channels 8-22

8.3.15 Restrictions in Access to PCI Bus ... 8-22
8.4 DMA Controller Registers ... 8-23
8.4.1 DMA Master Control Register (DM0MCR, DM1MCR) ... 8-25
8.4.2 DMA Channel Control Register (DM0CCRn, DM1CCRn) ... 8-27
8.4.3 DMA Channel Status Register (DM0CSRn, DM1CSRn)) ... 8-31
8.4.4 DMA Source Address Register (DM0SARn, DM1SARn) ... 8-33
8.4.5 DMA Destination Address Register (DM0DARn, DM1DARn) ... 8-34
8.4.6 DMA Chain Address Register (DM0CHARn, DM1CHARn) ... 8-35
8.4.7 DMA Source Address Increment Register (DM0SAIRn, DM1SAIRn) ... 8-36
8.4.8 DMA Destination Address Increment Register (DM0DAIRn, DM1DAIRn) ... 8-37
8.4.9 DMA Count Register (DM0CNTRn, DM1CNTRn) ... 8-38
8.4.10 DMA Memory Fiill Data Register (DM0MFDR, DM1MFDR) ... 8-39
8.5 Timing Diagrams ... 8-40
8.5.1 Single Address Single Transfer from Memory to I/O (32-bit ROM) ... 8-40
8.5.2 Single Address Single Transfer from Memory to I/O (16-bit ROM) ... 8-41
8.5.3 Single Address Single Transfer from I/O to Memory (32-bit SRAM) ... 8-42
8.5.4 Single Address Burst Transfer from Memory to I/O (32-bit ROM) ... 8-43
8.5.5 Single Address Burst Transfer from I/O to Memory (32-bit SRAM) ... 8-44
8.5.6 Single Address Single Transfer from Memory to I/O (16-bit ROM) ... 8-46
8.5.7 Single Address Single Transfer from I/O to Memory (16-bit SRAM) ... 8-47
8.5.8 Single Address Single Transfer from Memory to I/O (32-bit Half Speed ROM) ... 8-48
8.5.9 Single Address Single Transfer from I/O to Memory (32-bit Half Speed SRAM) ... 8-49
8.5.10 Single Address Single Transfer from Memory to I/O (64-bit SRAM) ... 8-50
8.5.11 Single Address Single Transfer from I/O to Memory (64-bit SDRAM) ... 8-51
8.5.12 Single Address Single Transfer from Memory to I/O of Last Cycle when DMADONE* Signal is Set to Output ... 8-52
8.5.13 Single Address Single Transfer from Memory to I/O (32-bit SDRAM) ... 8-53
8.5.14 Single Address Single Transfer from I/O to Memory (32-bit SDRAM) ... 8-54
8.5.15 External I/O Device – SRAM Dual Address Transfer ... 8-55
8.5.16 External I/O Device – SDRAM Dual Address Transfer ... 8-57
8.5.17 External I/O Device (Non-burst) – SDRAM Dual Address Transfer ... 8-59
9. SDRAM Controller ... 9-1
9.1 Characteristics ... 9-1
9.2 Block Diagram ... 9-2
9.3 Detailed Explanation ... 9-3
9.3.1 Supported SDRAM configurations ... 9-3
9.3.2 Address Mapping ... 9-4
9.3.3 Initialization of SDRAM ... 9-9
9.3.4 Initialization of Memory Data, ECC/Parity ... 9-10
9.3.5 Low Power Consumption Function ... 9-11
9.3.6 Bus Errors ... 9-12
9.3.7 Memory Read and Memory Write ... 9-12
9.3.8 Slow Write Burst ... 9-12
9.3.9 Clock Feedback ... 9-12
9.3.10 ECC ... 9-13
9.4 Registers ... 9-17
9.4.1 SDRAM Channel Control Register (SDCCRn) 0x8000 (ch. 0) 0x8008 (ch. 1) 0x8010 (ch. 2) 0x8018 (ch. 3) ... 9-18
9.4.2 SDRAM Timing Register (SDCTR) 0x8040 ... 9-20
9.4.3 SDRAM Command Register (SDCCMD) 0x8058 ... 9-22
9.4.4 ECC Control Register (ECCCR) 0xA000 ... 9-23
9.4.5 ECC Status Register (ECCSR) 0xA008 ... 9-25
9.5 Timing Diagrams ... 9-26
9.5.1 Single Read (64-bit Bus) ... 9-26
9.5.2 Single Write (64-bit Bus) ... 9-28
9.5.3 Burst Read (64-bit Bus) ... 9-30
9.5.4 Burst Write (64-bit Bus) ... 9-31

9.5.5 Burst Write (64-bit Bus, Slow Write Burst) ... 9-32
9.5.6 Single Read (32-bit Bus) ... 9-33
9.5.7 Single Write (32-bit Bus) ... 9-35
9.5.8 Low Power Consumption and Power Down Mode ... 9-37
9.6 SDRAM Usage Example ... 9-42
10. PCI Controller ... 10-1
10.1 Features ... 10-1
10.1.1 Overall ... 10-1
10.1.2 Initiator Function ... 10-1
10.1.3 Target Function ... 10-2
10.1.4 PCI Arbiter ... 10-2
10.1.5 PDMAC (PCI DMA Controller) ... 10-2
10.2 Block Diagram ... 10-3
10.3 Detailed Explanation ... 10-4
10.3.1 Terminology Explanation ... 10-4
10.3.2 On-chip Register ... 10-4
10.3.3 Supported PCI Bus Commands ... 10-6
10.3.4 Initiator Access (G-Bus → PCI Bus Address Conversion) ... 10-8
10.3.5 Target Access (PCI Bus → G-Bus Address Conversion) ... 10-10
10.3.6 Post Write Function ... 10-12
10.3.7 Endian Switching Function ... 10-12
10.3.8 66 MHz Operation Mode ... 10-13
10.3.9 Power Management ... 10-14
10.3.10 PDMAC (PCI DMA Controller) ... 10-15
10.3.11 Error Detection, Interrupt Reporting ... 10-18
10.3.12 PCI Bus Arbiter ... 10-20
10.3.13 PCI Boot ... 10-22
10.3.14 Set Configuration Space ... 10-23
10.3.15 PCI Clock ... 10-23
10.4 PCI Controller Control Register ... 10-24
10.4.1 ID Register (PCIID) 0xD000 ... 10-26
10.4.2 PCI Status, Command Register (PCISTATUS) 0xD004 ... 10-27
10.4.3 Class Code, Revision ID Register (PCICCREV) 0xD008 ... 10-30
10.4.4 PCI Configuration 1 Register (PCICFG1) 0xD00C ... 10-31
10.4.5 P2G Memory Space 0 PCI Lower Base Address Register (P2GM0PLBASE) 0xD010 ... 10-32
10.4.6 P2G Memory Space 0 PCI Upper Base Address Register (P2GM0PUBASE) 0xD014 ... 10-33
10.4.7 P2G Memory Space 1 PCI Lower Base Address Register (P2GM1PLBASE) 0xD018 ... 10-33
10.4.8 P2G Memory Space 1 PCI Upper Base Address Register (P2GM1PUBASE) 0xD01C ... 10-34
10.4.9 P2G Memory Space 2 PCI Base Address Register (P2GM2PBASE) 0xD020 ... 10-34
10.4.10 P2G I/O Space PCI Base Address Register (P2GIOPBASE) 0xD024 ... 10-35
10.4.11 Subsystem ID Register (PCISID) 0xD02C ... 10-36
10.4.12 Capabilities Pointer Register (PCICAPPTR) 0xD034 ... 10-37
10.4.13 PCI Configuration 2 Register (PCICFG2) 0xD03C ... 10-38
10.4.14 G2P Timeout Count Register (G2PTOCNT) 0xD040 ... 10-39
10.4.15 G2P Status Register (G2PSTATUS) 0xD080 ... 10-40
10.4.16 G2P Interrupt Mask Register (G2PMASK) 0xD084 ... 10-41
10.4.17 Satellite Mode PCI Status Register (PCISSTATUS) 0xD088 ... 10-42
10.4.18 PCI Status Interrupt Mask Register (PCIMASK) 0xD08C ... 10-43
10.4.19 P2G Configuration Register (P2GCFG) 0xD090 ... 10-44
10.4.20 P2G Status Register (P2GSTATUS) 0xD094 ... 10-46
10.4.21 P2G Interrupt Mask Register (P2GMASK) 0xD098 ... 10-47
10.4.22 P2G Current Command Register (P2GCCMD) 0xD09C ... 10-48
10.4.23 PCI Bus Arbiter Request Port Register (PBAREQPORT) 0xD100 ... 10-49
10.4.24 PCI Bus Arbiter Configuration Register (PBACFG) 0xD104 ... 10-51
10.4.25 PCI Bus Arbiter Status Register (PBASTATUS) 0xD108 ... 10-52
10.4.26 PCI Bus Arbiter Interrupt Mask Register (PBAMASK) 0xD10C ... 10-53
10.4.27 PCI Bus Arbiter Broken Master Register (PBABM) 0xD110 ... 10-54

10.4.28 PCI Bus Arbiter Current Request Register (PBACREQ) 0xD114 ... 10-55
10.4.29 PCI Bus Arbiter Current Grant Register (PBACGNT) 0xD118 ... 10-56
10.4.30 PCI Bus Arbiter Current State Register (PBACSTATE) 0xD11C ... 10-57
10.4.31 G2P Memory Space 0 G-Bus Base Address Register (G2PM0GBASE) 0xD120 ... 10-59
10.4.32 G2P Memory Space 1 G-Bus Base Address Register (G2PM1GBASE) 0xD128 ... 10-60
10.4.33 G2P Memory Space 2 G-Bus Base Address Register (G2PM2GBASE) 0xD130 ... 10-61
10.4.34 G2P I/O Space G-Bus Base Address Register (G2PIOGBASE) 0xD138 ... 10-62
10.4.35 G2P Memory Space 0 Address Mask Register (G2PM0MASK) 0xD140 ... 10-63
10.4.36 G2P Memory Space 1 Address Mask Register (G2PM1MASK) 0xD144 ... 10-64
10.4.37 G2P Memory Space 2 Address Mask Register (G2PM2MASK) 0xD148 ... 10-65
10.4.38 G2P I/O Space Address Mask Register (G2PIOMASK) 0xD14C ... 10-66
10.4.39 G2P Memory Space 0 PCI Base Address Register (G2PM0PBASE) 0xD150 ... 10-67
10.4.40 G2P Memory Space 1 PCI Base Address Register (G2PM1PBASE) 0xD158 ... 10-68
10.4.41 G2P Memory Space 2 PCI Base Address Register (G2PM2PBASE) 0xD160 ... 10-69
10.4.42 G2P I/O Space PCI Base Address Register (G2PIOPBASE) 0xD168 ... 10-70
10.4.43 PCI Controller Configuration Register (PCICCFG) 0xD170 ... 10-71
10.4.44 PCI Controller Status Register (PCICSTATUS) 0xD174 ... 10-74
10.4.45 PCI Controller Interrupt Mask Register (PCICMASK) 0xD178 ... 10-76
10.4.46 P2G Memory Space 0 G-Bus Base Address Register (P2GM0GBASE) 0xD180 ... 10-77
10.4.47 P2G Memory Space 1 G-Bus Base Address Register (P2GM1GBASE) 0xD188 ... 10-78
10.4.48 P2G Memory Space 2 G-Bus Base Address Register (P2GM2GBASE) 0xD190 ... 10-79
10.4.49 P2G I/O Space G-Bus Base Address Register (P2GIOGBASE) 0xD198 ... 10-80
10.4.50 G2P Configuration Address Register(G2PCFGADRS) 0xD1A0 ... 10-81
10.4.51 G2P Configuration Data Register (G2PCFGDATA) 0xD1A4 ... 10-82
10.4.52 G2P Interrupt Acknowledge Data Register (G2PINTACK) 0xD1C8 ... 10-83
10.4.53 G2P Special Cycle Data Register (G2PSPC) 0xD1CC ... 10-84
10.4.54 Configuration Data 0 Register (PCICDATA0) 0xD1D0 ... 10-85
10.4.55 Configuration Data 1 Register (PCICDATA1) 0xD1D4 ... 10-86
10.4.56 Configuration Data 2 Register (PCICDATA2) 0xD1D8 ... 10-87
10.4.57 Configuration Data 3 Register (PCICDATA3) 0xD1DC ... 10-88
10.4.58 PDMAC Chain Address Register (PDMCA) 0xD200 ... 10-89
10.4.59 PDMAC G-Bus Address Register (PDMGA) 0xD208 ... 10-90
10.4.60 PDMAC PCI Bus Address Register (PDMPA) 0xD210 ... 10-91
10.4.61 PDMAC Count Register (PDMCTR) 0xD218 ... 10-92
10.4.62 PDMAC Configuration Register (PDMCFG) 0xD220 ... 10-93
10.4.63 PDMAC Status Register (PDMSTATUS) 0xD228 ... 10-95
10.5 PCI Configuration Space Register ... 10-98
10.5.1 Capability ID Register (Cap_ID) 0xDC ... 10-99
10.5.2 Next Item Pointer Register (Next_Item_Ptr) 0xDD ... 10-100
10.5.3 Power Management Capability Register (PMC) 0xDE ... 10-101
10.5.4 Power Management Control/Status Register (PMCSR) 0xE0 ... 10-102

11. Serial I/O Port ... 11-1
11.1 Features ... 11-1
11.2 Block Diagram ... 11-2
11.3 Detailed Explanation ... 11-3
11.3.1 Overview ... 11-3
11.3.2 Data Format ... 11-3
11.3.3 Serial Clock Generator ... 11-5
11.3.4 Data Reception ... 11-7
11.3.5 Data Transmission ... 11-7
11.3.6 DMA Transfer ... 11-7
11.3.7 Flow Control ... 11-8
11.3.8 Reception Data Status ... 11-8
11.3.9 Reception Time Out ... 11-9
11.3.10 Software Reset ... 11-9
11.3.11 Error Detection/Interrupt Signaling ... 11-10
11.3.12 Multi-Controller System ... 11-11
11.4 Registers ... 11-12

11.4.1 Line Control Register 0 (SILCR0) 0xF300 (Ch. 0)
Line Control Register 1 (SILCR1) 0xF400 (Ch. 1) 11-13
11.4.2 DMA/Interrupt Control Register 0 (SIDICR0) 0xF304 (Ch. 0)
DMA/Interrupt Control Register 1 (SIDICR1) 0xF404 (Ch. 1)....11-15
11.4.3 DMA/Interrupt Status Register 0 (SIDISR0) 0xF308 (Ch. 0)
DMA/Interrupt Status Register 1 (SIDISR1) 0xF408 (Ch. 1) 11-17
11.4.4 Status Change Interrupt Status Register 0 (SISCISR0) 0xF30C (Ch. 0)
Status Change Interrupt Status Register 1 (SISCISR1) 0xF40C (Ch. 1) 11-19
11.4.5 FIFO Control Register 0 (SIFCR0) 0xF310 (Ch. 0)
FIFO Control Register 1 (SIFCR1) 0xF410 (Ch. 1) 11-20
11.4.6 Flow Control Register 0 (SIFLCR0) 0xF314 (Ch. 0)
Flow Control Register 1 (SIFLCR1) 0xF414 (Ch. 1) 11-21
11.4.7 Baud Rate Control Register 0 (SIBGR0) 0xF318 (Ch. 0)
Baud Rate Control Register 1 (SIBGR1) 0xF418 (Ch. 1) 11-22
11.4.8 Transmit FIFO Register 0 (SITFIFO0) 0xF31C (Ch. 0)
Transmit FIFO Register 1 (SITFIFO1) 0xF41C (Ch. 1)....11-23
11.4.9 Receive FIFO Register 0 (SIRFIFO0) 0xF320 (Ch. 0)
Receive FIFO Register 1 (SIRFIFO1) 0xF420 (Ch. 1)....11-24

12. Timer/Counter 12-1
12.1 Features 12-1
12.2 Block Diagram 12-2
12.3 Detailed Explanation 12-3
12.3.1 Overview 12-3
12.3.2 Counter Clock 12-3
12.3.3 Counter 12-4
12.3.4 Interval Timer Mode 12-4
12.3.5 Pulse Generator Mode 12-6
12.3.6 Watchdog Timer Mode 12-7
12.4 Registers 12-9
12.4.1 Timer Control Register *n* (TMTCRn) TMTCR0 0xF000 TMTCR1 0xF100 TMTCR2 0xF200 12-10
12.4.2 Timer Interrupt Status Register *n* (TMTISRn) TMTISR0 0xF004 TMTISR1 0xF104
TMTISR2 0xF204 12-11
12.4.3 Compare Register An (TMCPRAn) TMCPRA0 0xF008 TMCPRA1 0xF108
TMCPRA2 0xF208 12-12
12.4.4 Compare Register Bn (TMCPRBn) TMCPRB0 0xF00C TMCPRB1 0xF10C 12-13
12.4.5 Interval Timer Mode Register *n* (TMITMRn) TMITMR0 0xF010 TMITMR1 0xF110
TMITMR2 0xF210 12-14
12.4.6 Divide Register *n* (TMCCDRn) TMCCDR0 0xF020 TMCCDR1 0xF120 TMCCDR2 0xF220 .. 12-15
12.4.7 Pulse Generator Mode Register *n* (TMPGMRn) TMPGMR0 0xF000 TMPGMR1 0xF130 12-16
12.4.8 Watchdog Timer Mode Register *n* (TMWTMRn) TMWTMR2 0xF240 12-17
12.4.9 Timer Read Register *n* (TMTRRn) 0xF0F0 TMTRR0 0xF0F0 TMTRR1 0xF1F0
TMTRR2 0xF2F0 12-18

13. Parallel I/O Port 13-1
13.1 Characteristics 13-1
13.2 Block Diagram 13-1
13.3 Detailed Description 13-2
13.3.1 Selecting PIO Pins 13-2
13.3.2 General-purpose Parallel Port 13-2
13.4 Registers 13-2
13.4.1 PIO Output Data Register (PIODO) 0xF500 13-3
13.4.2 PIO Input Data Register (PIODI) 0xF504 13-3
13.4.3 PIO Direction Control Register (PIODIR) 0xF508 13-4
13.4.4 PIO Open Drain Control Register (XPIOOD) 0xF50C 13-4

14. AC-link Controller 14-1
14.1 Features 14-1

14.2 Configuration 14-2
14.3 Functional Description 14-3
14.3.1 CODEC Connection 14-3
14.3.2 Boot Configuration 14-4
14.3.3 Usage Flow 14-5
14.3.4 AC-link Start Up 14-7
14.3.5 CODEC Register Access 14-8
14.3.6 Sample-data Transmission and Reception 14-9
14.3.7 GPIO Operation 14-14
14.3.8 Interrupt 14-15
14.3.9 AC-link Low-power Mode 14-15
14.4 Registers 14-16
14.4.1 ACLC Control Enable Register 0xF700 14-17
14.4.2 ACLC Control Disable Register 0xF704 14-20
14.4.3 ACLC CODEC Register Access Register 0xF708 14-22
14.4.4 ACLC Interrupt Status Register 0xF710 14-23
14.4.5 ACLC Interrupt Masked Status Register 0xF714 14-25
14.4.6 ACLC Interrupt Enable Register 0xF718 14-25
14.4.7 ACLC Interrupt Disable Register 0xF71C 14-25
14.4.8 ACLC Semaphore Register 0xF720 14-26
14.4.9 ACLC GPI Data Register 0xF740 14-27
14.4.10 ACLC GPO Data Register 0xF744 14-28
14.4.11 ACLC Slot Enable Register 0xF748 14-29
14.4.12 ACLC Slot Disable Register 0xF74C 14-31
14.4.13 ACLC FIFO Status Register 0xF750 14-32
14.4.14 ACLC DMA Request Status Register 0xF780 14-34
14.4.15 ACLC DMA Channel Selection Register 0xF784 14-35
14.4.16 ACLC Audio PCM Output Data Register 0xF7A0 14-36
14.4.17 ACLC Center Data Register 0xF7A8 14-37
14.4.18 ACLC Audio PCM Input Data Register 0xF7B0 14-38
14.4.19 ACLC Modem Input Data Register 0xF7BC 14-39
14.4.20 ACLC Revision ID Register 0xF7FC 14-40

15. Interrupt Controller 15-1
15.1 Characteristics 15-1
15.2 Block Diagram 15-2
15.3 Detailed Explanation 15-4
15.3.1 Interrupt sources 15-4
15.3.2 Interrupt request detection 15-5
15.3.3 Interrupt level assigning 15-5
15.3.4 Interrupt priority assigning 15-6
15.3.5 Interrupt notification 15-7
15.3.6 Clearing interrupt requests 15-7
15.3.7 Interrupt requests 15-8
15.4 Registers 15-10
15.4.1 Interrupt Detection Enable Register (IRDEN) 0xF600 15-11
15.4.2 Interrupt Detection Mode Register 0 (IRDM0) 0xF604 15-12
15.4.3 Interrupt Detection Mode Register 1 (IRDM1) 0xF608 15-14
15.4.4 Interrupt Level Register 0 (IRLVL0) 0xF610 15-17
15.4.5 Interrupt Level Register (IRLVL1) 0xF614 15-19
15.4.6 Interrupt Level Register 2 (IRLVL2) 0xF618 15-21
15.4.7 Interrupt Level Register 3 (IRLVL3) 0xF61C 15-22
15.4.8 Interrupt Level Register 4 (IRLVL4) 0xF620 15-24
15.4.9 Interrupt Level Register 5 (IRLVL5) 0xF624 15-26
15.4.10 Interrupt Level Register 6 (IRLVL6) 0xF628 15-28
15.4.11 Interrupt Level Register 7 (IRLVL7) 0xF62C 15-30
15.4.12 Interrupt Mask Level Register (IRMSK) 0xF640 15-32
15.4.13 Interrupt Edge Detection Clear Register (IREDC) 0xF660 15-33

15.4.14 Interrupt Pending Register (IRPND) 0xF680 ... 15-34
15.4.15 Interrupt Current Status Register (IRCS) 0xF6A0 ... 15-37
15.4.16 Interrupt Request Flag Register 0 (IRFLAG0) 0xF510 ... 15-39
15.4.17 Interrupt Request Flag Register 1 (IRFLAG1) 0xF514 ... 15-40
15.4.18 Interrupt Request Polarity Control Register (IRPOL) 0xF518 ... 15-41
15.4.19 Interrupt Request Control Register (IRRCNT) 0xF51C ... 15-42
15.4.20 Interrupt Request Internal Interrupt Mask Register (IRMASKINT) 0xF520 ... 15-43
15.4.21 Interrupt Request External Interrupt Mask Register (IRMASKEXT) 0xF524 ... 15-44

16. Ethernet Controller ... 16-1
16.1 Features ... 16-1
16.2 Block diagram ... 16-1
16.3 Detailed explanation ... 16-2
16.3.1 Accessing the Ethernet Controller ... 16-2
16.3.2 Data structure ... 16-4
16.3.3 System control model ... 16-5
16.3.4 Functional overview ... 16-6
16.3.5 DMA function block ... 16-10
16.3.6 MAC function blocks ... 16-11
16.3.7 Memory configuration ... 16-13
16.3.8 MAC operation ... 16-20
16.3.9 DMA operation ... 16-32
16.4 Registers ... 16-37
16.4.1 Overview ... 16-37
16.4.2 PCI Configuration Register group ... 16-42
16.4.3 DMA Control, Status Register group ... 16-54
16.4.4 Flow Control Register group ... 16-68
16.4.5 MAC Control, Status Register group ... 16-70

17. SPI (Serial Peripheral Interface) Module (SPI) ... 17-1
17.1 Characteristics ... 17-1
17.2 Block diagram ... 17-2
17.3 Operational description ... 17-3
17.3.1 Operation modes ... 17-3
17.3.2 Transmitter/Receiver ... 17-3
17.3.3 Baud Rate Generator ... 17-4
17.3.4 Transfer format ... 17-5
17.3.5 Interframe Delay Time Counter ... 17-6
17.3.6 Buffer configuration ... 17-7
17.3.7 SPI system errors ... 17-7
17.3.8 Interrupts ... 17-7
17.4 Registers ... 17-8
17.4.1 SPI Master Control Register (SPMCR) 0xF800 ... 17-8
17.4.2 SPI Control Register 0 (SPCR0) 0xF804 ... 17-9
17.4.3 SPI Control Register 1 (SPCR1) 0xF808 ... 17-11
17.4.4 SPI Interframe Delay Time Counter (SPFS) 0xF80C ... 17-12
17.4.5 SPI Status Register (SPSR) 0xF814 ... 17-13
17.4.6 SPI Data Register (SPDR) 0xF818 ... 17-14

18. NAND Flash Memory IPL ... 18-1
18.1 Features ... 18-1
18.2 Block Diagram ... 18-2
18.2.1 System Block Diagram ... 18-2
18.2.2 Function Block Diagram ... 18-2
18.3 Detailed Explanation ... 18-3
18.3.1 NAND IPL Operation Conditions ... 18-3
18.3.2 NAND IPL Operation ... 18-4
18.3.3 NAND Flash Memory IPL Address Map ... 18-8

18.3.4 NAND Flash Memory IPL Control Information....18-11
18.3.5 Data configuration of NAND Flash Memory....18-17
18.3.6 NAND Flash Memory IPL Error Handling....18-26

19. On-Chip SRAM....19-1
19.1 Characteristics....19-1
19.2 Block diagram....19-1
19.3 Detailed explanation....19-2
19.3.1 Base address....19-2
19.3.2 Access cycle count....19-2
19.4 Register....19-3
19.4.1 On-chip SRAM Control Register (SRAMCR : 0x6000)....19-3

20. NAND Flash Memory Controller....20-1
20.1 Features....20-1
20.2 Block diagram....20-1
20.3 Detailed operation....20-2
20.3.1 Accessing NAND flash memory....20-2
20.3.2 ECC control....20-5
20.4 Registers....20-6
20.4.1 NAND Flash Memory Data Transfer Register (NDFDTR) 0x5000....20-6
20.4.2 NAND Flash Memory Mode Control Register (NDFMCR) 0x5008....20-7
20.4.3 NAND Flash Memory Status Register (NDFSR) 0x5010....20-8
20.4.4 NAND Flash Memory Interrupt Status Register (NDFISR) 0x5018....20-9
20.4.5 NAND Flash Memory Interrupt Mask Register (NDFIMR) 0x5020....20-10
20.4.6 NAND Flash Memory Strobe Pulse Width Register (NDFSPR) 0x5028....20-11
20.4.7 NAND Flash Memory Reset Register (NDFRSTR) 0x5030....20-12
20.5 Timing diagram....20-13
20.5.1 Command cycles and address cycles....20-13
20.5.2 Data read cycles....20-14
20.5.3 Data write cycles....20-15
20.6 NAND flash memory connection example....20-16

21. Extended EJTAG Interface....21-1
21.1 Extended EJTAG Interface....21-1
21.2 JTAG Boundary Scan Test....21-2
21.2.1 JTAG Controller and Register....21-2
21.2.2 Instruction Register....21-3
21.2.3 Boundary Scan Register....21-3
21.2.4 Device ID Register....21-6
21.3 Initializing the Extended EJTAG Interface....21-7

22. Electrical Characteristics....22-1
22.1 Absolute maximum rating (*1)....22-1
22.2 Recommended operating conditions....22-1
22.3 DC characteristics....22-2
22.3.1 DC characteristics of pins other than PCI Interface pins....22-2
22.3.2 DC characteristics of PCI Interface pins....22-3
22.4 PLL power....22-4
22.4.1 PLL power connection example....22-4
22.5 AC characteristics....22-5
22.5.1 MASTERCLK AC characteristics....22-5
22.5.2 Power on AC characteristics....22-5
22.5.3 SDRAM Interface AC characteristics....22-6
22.5.4 External Bus Interface AC characteristics....22-8
22.5.5 PCI Interface AC characteristics (66 MHz)....22-9
22.5.6 PCI Interface AC characteristics (33 MHz)....22-9

22.5.7 PCI EEPROM Interface AC characteristics 22-11
22.5.8 DMA Interface AC characteristics 22-11
22.5.9 Interrupt Interface AC characteristics 22-12
22.5.10 SIO Interface AC characteristics 22-13
22.5.11 Timer Interface AC characteristics 22-13
22.5.12 PIO Interface AC characteristics 22-14
22.5.13 AC-link Interface AC characteristics 22-14
22.5.14 NAND Flash Memory Interface AC characteristics 22-15
22.5.15 SPI AC characteristics 22-16
22.5.16 Ethernet Interface (MII) AC characteristics 22-17

23. Pinout and Package Information 23-1
23.1 Pinout Diagram 23-1
23.2 Package Dimensions 23-9

24. Notes on Use of TMPR4938 24-1
24.1 Notes on TX49/H3 Core 24-1
24.2 Notes on External Bus Controller 24-3
24.3 Notes on DMA Controller 24-3
24.4 Note on PCI Controller 24-6
24.5 Notes on Serial I/O Port 24-10
24.6 Notes on Ether Controller 24-11

25. Parts Number when Ordering 25-1

Appendix A TX49/H3 Core Supplement A-1
A.1 Processor ID A-1
A.2 Interrupts A-1
A.3 Bus Snoop A-1
A.4 Halt/Doze mode A-1
A.5 Memory access order A-1

TMPR4938 Revision History 1

Handling Precautions

1. Using Toshiba Semiconductors Safely

TOSHIBA are continually working to improve the quality and the reliability of their products.

Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property.

In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.

The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury (“Unintended Usage”). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer’s own risk.

2. Safety Precautions

This section lists important precautions which users of semiconductor devices (and anyone else) should observe in order to avoid injury and damage to property, and to ensure safe and correct use of devices.

Please be sure that you understand the meanings of the labels and the graphic symbol described below before you move on to the detailed descriptions of the precautions.

[Explanation of labels]

⚠DANGER	Indicates an imminently hazardous situation which will result in death or serious injury if you do not follow instructions.
⚠WARNING	Indicates a potentially hazardous situation which could result in death or serious injury if you do not follow instructions.
⚠CAUTION	Indicates a potentially hazardous situation which if not avoided, may result in minor injury or moderate injury.

[Explanation of graphic symbol]

Graphic symbol	Meaning
	Indicates that caution is required (laser beam is dangerous to eyes).

2.1 General Precautions regarding Semiconductor Devices

⚠CAUTION
Do not use devices under conditions exceeding their absolute maximum ratings (e.g. current, voltage, power dissipation or temperature). This may cause the device to break down, degrade its performance, or cause it to catch fire or explode resulting in injury. Do not insert devices in the wrong orientation. Make sure that the positive and negative terminals of power supplies are connected correctly. Otherwise the rated maximum current or power dissipation may be exceeded and the device may break down or undergo performance degradation, causing it to catch fire or explode and resulting in injury. When power to a device is on, do not touch the device's heat sink. Heat sinks become hot, so you may burn your hand. Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger. When conducting any kind of evaluation, inspection or testing, be sure to connect the testing equipment's electrodes or probes to the pins of the device under test before powering it on. Otherwise, you may receive an electric shock causing injury. Before grounding an item of measuring equipment or a soldering iron, check that there is no electrical leakage from it. Electrical leakage may cause the device which you are testing or soldering to break down, or could give you an electric shock. Always wear protective glasses when cutting the leads of a device with clippers or a similar tool. If you do not, small bits of metal flying off the cut ends may damage your eyes.

2.2 Precautions Specific to Each Product Group

2.2.1 Optical semiconductor devices

⚠ DANGER

When a visible semiconductor laser is operating, do not look directly into the laser beam or look through the optical system. This is highly likely to impair vision, and in the worst case may cause blindness.
If it is necessary to examine the laser apparatus, for example to inspect its optical characteristics, always wear the appropriate type of laser protective glasses as stipulated by IEC standard IEC825-1.

⚠ WARNING

Ensure that the current flowing in an LED device does not exceed the device's maximum rated current.
This is particularly important for resin-packaged LED devices, as excessive current may cause the package resin to blow up, scattering resin fragments and causing injury.

When testing the dielectric strength of a photocoupler, use testing equipment which can shut off the supply voltage to the photocoupler. If you detect a leakage current of more than 100 μA, use the testing equipment to shut off the photocoupler's supply voltage; otherwise a large short-circuit current will flow continuously, and the device may break down or burst into flames, resulting in fire or injury.

When incorporating a visible semiconductor laser into a design, use the device's internal photodetector or a separate photodetector to stabilize the laser's radiant power so as to ensure that laser beams exceeding the laser's rated radiant power cannot be emitted.
If this stabilizing mechanism does not work and the rated radiant power is exceeded, the device may break down or the excessively powerful laser beams may cause injury.

2.2.2 Power devices

⚠ DANGER

Never touch a power device while it is powered on. Also, after turning off a power device, do not touch it until it has thoroughly discharged all remaining electrical charge.
Touching a power device while it is powered on or still charged could cause a severe electric shock, resulting in death or serious injury.

When conducting any kind of evaluation, inspection or testing, be sure to connect the testing equipment's electrodes or probes to the device under test before powering it on.
When you have finished, discharge any electrical charge remaining in the device.
Connecting the electrodes or probes of testing equipment to a device while it is powered on may result in electric shock, causing injury.

⚠WARNING

Do not use devices under conditions which exceed their absolute maximum ratings (current, voltage, power dissipation, temperature etc.).
This may cause the device to break down, causing a large short-circuit current to flow, which may in turn cause it to catch fire or explode, resulting in fire or injury.

Use a unit which can detect short-circuit currents and which will shut off the power supply if a short-circuit occurs.
If the power supply is not shut off, a large short-circuit current will flow continuously, which may in turn cause the device to catch fire or explode, resulting in fire or injury.

When designing a case for enclosing your system, consider how best to protect the user from shrapnel in the event of the device catching fire or exploding.
Flying shrapnel can cause injury.

When conducting any kind of evaluation, inspection or testing, always use protective safety tools such as a cover for the device.
Otherwise you may sustain injury caused by the device catching fire or exploding.

Make sure that all metal casings in your design are grounded to earth.
Even in modules where a device's electrodes and metal casing are insulated, capacitance in the module may cause the electrostatic potential in the casing to rise.
Dielectric breakdown may cause a high voltage to be applied to the casing, causing electric shock and injury to anyone touching it.

When designing the heat radiation and safety features of a system incorporating high-speed rectifiers, remember to take the device's forward and reverse losses into account.
The leakage current in these devices is greater than that in ordinary rectifiers; as a result, if a high-speed rectifier is used in an extreme environment (e.g. at high temperature or high voltage), its reverse loss may increase, causing thermal runaway to occur.
This may in turn cause the device to explode and scatter shrapnel, resulting in injury to the user.

A design should ensure that, except when the main circuit of the device is active, reverse bias is applied to the device gate while electricity is conducted to control circuits, so that the main circuit will become inactive.
Malfunction of the device may cause serious accidents or injuries.

⚠CAUTION

When conducting any kind of evaluation, inspection or testing, either wear protective gloves or wait until the device has cooled properly before handling it.
Devices become hot when they are operated. Even after the power has been turned off, the device will retain residual heat which may cause a burn to anyone touching it.

2.2.3 Bipolar ICs (for use in automobiles)

⚠CAUTION

If your design includes an inductive load such as a motor coil, incorporate diodes or similar devices into the design to prevent negative current from flowing in.
The load current generated by powering the device on and off may cause it to function erratically or to break down, which could in turn cause injury.

Ensure that the power supply to any device which incorporates protective functions is stable.
If the power supply is unstable, the device may operate erratically, preventing the protective functions from working correctly. If protective functions fail, the device may break down causing injury to the user.

3. General Safety Precautions and Usage Considerations

This section is designed to help you gain a better understanding of semiconductor devices, so as to ensure the safety, quality and reliability of the devices which you incorporate into your designs.

3.1 From Incoming to Shipping

3.1.1 Electrostatic discharge (ESD)

When handling individual devices (which are not yet mounted on a printed circuit board), be sure that the environment is protected against electrostatic electricity. Operators should wear anti-static clothing, and containers and other objects which come into direct contact with devices should be made of anti-static materials and should be grounded to earth via an 0.5- to 1.0-MΩ protective resistor.

Please follow the precautions described below; this is particularly important for devices which are marked “Be careful of static.”.

(1) Work environment

- When humidity in the working environment decreases, the human body and other insulators can easily become charged with static electricity due to friction. Maintain the recommended humidity of 40% to 60% in the work environment, while also taking into account the fact that moisture-proof-packed products may absorb moisture after unpacking.

- Be sure that all equipment, jigs and tools in the working area are grounded to earth.

- Place a conductive mat over the floor of the work area, or take other appropriate measures, so that the floor surface is protected against static electricity and is grounded to earth. The surface resistivity should be 10^4 to 10^8 Ω/sq and the resistance between surface and ground, 7.5×10^5 to 10^8 Ω

- Cover the workbench surface also with a conductive mat (with a surface resistivity of 10^4 to 10^8 Ω/sq, for a resistance between surface and ground of 7.5×10^5 to 10^8 Ω) . The purpose of this is to disperse static electricity on the surface (through resistive components) and ground it to earth. Workbench surfaces must not be constructed of low-resistance metallic materials that allow rapid static discharge when a charged device touches them directly.

- Pay attention to the following points when using automatic equipment in your workplace:

  (a) When picking up ICs with a vacuum unit, use a conductive rubber fitting on the end of the pick-up wand to protect against electrostatic charge.

  (b) Minimize friction on IC package surfaces. If some rubbing is unavoidable due to the device’s mechanical structure, minimize the friction plane or use material with a small friction coefficient and low electrical resistance. Also, consider the use of an ionizer.

  (c) In sections which come into contact with device lead terminals, use a material which dissipates static electricity.

  (d) Ensure that no statically charged bodies (such as work clothes or the human body) touch the devices.

(e) Make sure that sections of the tape carrier which come into contact with installation devices or other electrical machinery are made of a low-resistance material.

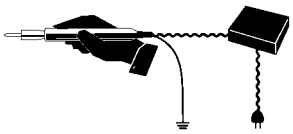
(f) Make sure that jigs and tools used in the assembly process do not touch devices.

(g) In processes in which packages may retain an electrostatic charge, use an ionizer to neutralize the ions.

- Make sure that CRT displays in the working area are protected against static charge, for example by a VDT filter. As much as possible, avoid turning displays on and off. Doing so can cause electrostatic induction in devices.
- Keep track of charged potential in the working area by taking periodic measurements.
- Ensure that work chairs are protected by an anti-static textile cover and are grounded to the floor surface by a grounding chain. (Suggested resistance between the seat surface and grounding chain is 7.5×10^5 to $10^{12}\Omega$.)
- Install anti-static mats on storage shelf surfaces. (Suggested surface resistivity is 10^4 to 10^8 Ω/sq; suggested resistance between surface and ground is 7.5×10^5 to 10^8 Ω.)
- For transport and temporary storage of devices, use containers (boxes, jigs or bags) that are made of anti-static materials or materials which dissipate electrostatic charge.
- Make sure that cart surfaces which come into contact with device packaging are made of materials which will conduct static electricity, and verify that they are grounded to the floor surface via a grounding chain.
- In any location where the level of static electricity is to be closely controlled, the ground resistance level should be Class 3 or above. Use different ground wires for all items of equipment which may come into physical contact with devices.

(2) Operating environment

- Operators must wear anti-static clothing and conductive shoes (or a leg or heel strap).
- Operators must wear a wrist strap grounded to earth via a resistor of about 1 MΩ.

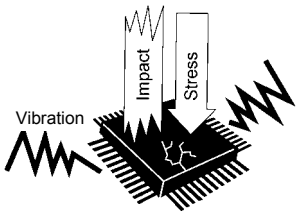


- Soldering irons must be grounded from iron tip to earth, and must be used only at low voltages (6 V to 24 V).
- If the tweezers you use are likely to touch the device terminals, use anti-static tweezers and in particular avoid metallic tweezers. If a charged device touches a low-resistance tool, rapid discharge can occur. When using vacuum tweezers, attach a conductive chucking pat to the tip, and connect it to a dedicated ground used especially for anti-static purposes (suggested resistance value: 10^4 to 10^8 Ω).
- Do not place devices or their containers near sources of strong electrical fields (such as above a CRT).

- When storing printed circuit boards which have devices mounted on them, use a board container or bag that is protected against static charge. To avoid the occurrence of static charge or discharge due to friction, keep the boards separate from one other and do not stack them directly on top of one another.
- Ensure, if possible, that any articles (such as clipboards) which are brought to any location where the level of static electricity must be closely controlled are constructed of anti-static materials.
- In cases where the human body comes into direct contact with a device, be sure to wear anti-static finger covers or gloves (suggested resistance value: 10^8 Ω or less).
- Equipment safety covers installed near devices should have resistance ratings of 10^9 Ω or less.
- If a wrist strap cannot be used for some reason, and there is a possibility of imparting friction to devices, use an ionizer.
- The transport film used in TCP products is manufactured from materials in which static charges tend to build up. When using these products, install an ionizer to prevent the film from being charged with static electricity. Also, ensure that no static electricity will be applied to the product's copper foils by taking measures to prevent static occuring in the peripheral equipment.

3.1.2 Vibration, impact and stress

Handle devices and packaging materials with care. To avoid damage to devices, do not toss or drop packages. Ensure that devices are not subjected to mechanical vibration or shock during transportation. Ceramic package devices and devices in canister-type packages which have empty space inside them are subject to damage from vibration and shock because the bonding wires are secured only at their ends.



Plastic molded devices, on the other hand, have a relatively high level of resistance to vibration and mechanical shock because their bonding wires are enveloped and fixed in resin. However, when any device or package type is installed in target equipment, it is to some extent susceptible to wiring disconnections and other damage from vibration, shock and stressed solder junctions. Therefore when devices are incorporated into the design of equipment which will be subject to vibration, the structural design of the equipment must be thought out carefully.

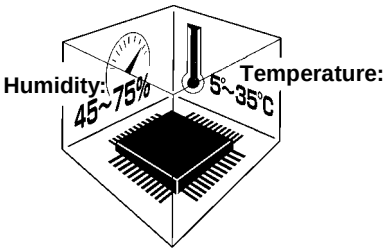
If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack. In products such as CCDs which incorporate window glass, this could cause surface flaws in the glass or cause the connection between the glass and the ceramic to separate.

Furthermore, it is known that stress applied to a semiconductor device through the package changes the resistance characteristics of the chip because of piezoelectric effects. In analog circuit design attention must be paid to the problem of package stress as well as to the dangers of vibration and shock as described above.

3.2 Storage

3.2.1 General storage

- Avoid storage locations where devices will be exposed to moisture or direct sunlight.
- Follow the instructions printed on the device cartons regarding transportation and storage.
- The storage area temperature should be kept within a temperature range of 5°C to 35°C, and relative humidity should be maintained at between 45% and 75%.



- Do not store devices in the presence of harmful (especially corrosive) gases, or in dusty conditions.
- Use storage areas where there is minimal temperature fluctuation. Rapid temperature changes can cause moisture to form on stored devices, resulting in lead oxidation or corrosion. As a result, the solderability of the leads will be degraded.
- When repacking devices, use anti-static containers.
- Do not allow external forces or loads to be applied to devices while they are in storage.
- If devices have been stored for more than two years, their electrical characteristics should be tested and their leads should be tested for ease of soldering before they are used.

3.2.2 Moisture-proof packing

Moisture-proof packing should be handled with care. The handling procedure specified for each packing type should be followed scrupulously. If the proper procedures are not followed, the quality and reliability of devices may be degraded. This section describes general precautions for handling moisture-proof packing. Since the details may differ from device to device, refer also to the relevant individual datasheets or databook.

(1) General precautions
Follow the instructions printed on the device cartons regarding transportation and storage.

- Do not drop or toss device packing. The laminated aluminum material in it can be rendered ineffective by rough handling.
- The storage area temperature should be kept within a temperature range of 5°C to 30°C, and relative humidity should be maintained at 90% (max). Use devices within 12 months of the date marked on the package seal.

- If the 12-month storage period has expired, or if the 30% humidity indicator shown in Figure 1 is pink when the packing is opened, it may be advisable, depending on the device and packing type, to back the devices at high temperature to remove any moisture. Please refer to the table below. After the pack has been opened, use the devices in a 5°C to 30°C. 60% RH environment and within the effective usage period listed on the moisture-proof package. If the effective usage period has expired, or if the packing has been stored in a high-humidity environment, bake the devices at high temperature.

Packing	Moisture removal
Tray	If the packing bears the "Heatproof" marking or indicates the maximum temperature which it can withstand, bake at 125°C for 20 hours. (Some devices require a different procedure.)
Tube	Transfer devices to trays bearing the "Heatproof" marking or indicating the temperature which they can withstand, or to aluminum tubes before baking at 125°C for 20 hours.
Tape	Deviced packed on tape cannot be baked and must be used within the effective usage period after unpacking, as specified on the packing.

- When baking devices, protect the devices from static electricity.

- Moisture indicators can detect the approximate humidity level at a standard temperature of 25°C. 6-point indicators and 3-point indicators are currently in use, but eventually all indicators will be 3-point indicators.

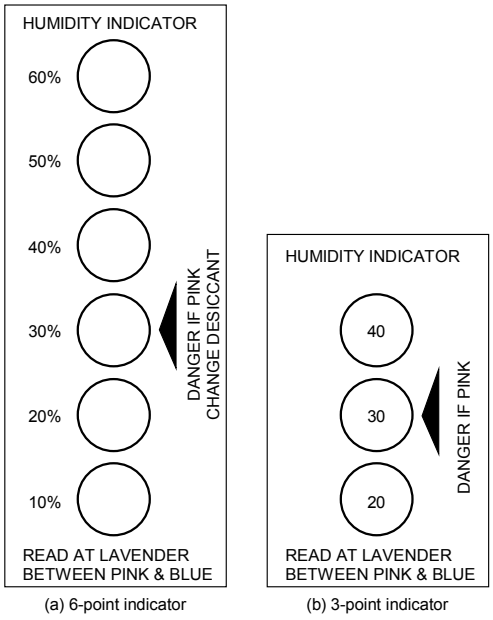


Figure 1 Humidity indicator

3.3 Design

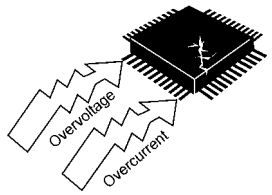
Care must be exercised in the design of electronic equipment to achieve the desired reliability. It is important not only to adhere to specifications concerning absolute maximum ratings and recommended operating conditions, it is also important to consider the overall environment in which equipment will be used, including factors such as the ambient temperature, transient noise and voltage and current surges, as well as mounting conditions which affect device reliability. This section describes some general precautions which you should observe when designing circuits and when mounting devices on printed circuit boards.

For more detailed information about each product family, refer to the relevant individual technical datasheets available from Toshiba.

3.3.1 Absolute maximum ratings

⚠CAUTION Do not use devices under conditions in which their absolute maximum ratings (e.g. current, voltage, power dissipation or temperature) will be exceeded. A device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user.

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Although absolute maximum ratings differ from product to product, they essentially concern the voltage and current at each pin, the allowable power dissipation, and the junction and storage temperatures.



If the voltage or current on any pin exceeds the absolute maximum rating, the device's internal circuitry can become degraded. In the worst case, heat generated in internal circuitry can fuse wiring or cause the semiconductor chip to break down.

If storage or operating temperatures exceed rated values, the package seal can deteriorate or the wires can become disconnected due to the differences between the thermal expansion coefficients of the materials from which the device is constructed.

3.3.2 Recommended operating conditions

The recommended operating conditions for each device are those necessary to guarantee that the device will operate as specified in the datasheet.
If greater reliability is required, derate the device's absolute maximum ratings for voltage, current, power and temperature before using it.

3.3.3 Derating

When incorporating a device into your design, reduce its rated absolute maximum voltage, current, power dissipation and operating temperature in order to ensure high reliability.
Since derating differs from application to application, refer to the technical datasheets available for the various devices used in your design.

3.3.4 Unused pins

If unused pins are left open, some devices can exhibit input instability problems, resulting in malfunctions such as abrupt increase in current flow. Similarly, if the unused output pins on a device are connected to the power supply pin, the ground pin or to other output pins, the IC may malfunction or break down.

Since the details regarding the handling of unused pins differ from device to device and from pin to pin, please follow the instructions given in the relevant individual datasheets or databook.

CMOS logic IC inputs, for example, have extremely high impedance. If an input pin is left open, it can easily pick up extraneous noise and become unstable. In this case, if the input voltage level reaches an intermediate level, it is possible that both the P-channel and N-channel transistors will be turned on, allowing unwanted supply current to flow. Therefore, ensure that the unused input pins of a device are connected to the power supply (Vcc) pin or ground (GND) pin of the same device. For details of what to do with the pins of heat sinks, refer to the relevant technical datasheet and databook.

3.3.5 Latch-up

Latch-up is an abnormal condition inherent in CMOS devices, in which Vcc gets shorted to ground. This happens when a parasitic PN-PN junction (thyristor structure) internal to the CMOS chip is turned on, causing a large current of the order of several hundred mA or more to flow between Vcc and GND, eventually causing the device to break down.

Latch-up occurs when the input or output voltage exceeds the rated value, causing a large current to flow in the internal chip, or when the voltage on the Vcc (Vdd) pin exceeds its rated value, forcing the internal chip into a breakdown condition. Once the chip falls into the latch-up state, even though the excess voltage may have been applied only for an instant, the large current continues to flow between Vcc (Vdd) and GND (Vss). This causes the device to heat up and, in extreme cases, to emit gas fumes as well. To avoid this problem, observe the following precautions:

(1) Do not allow voltage levels on the input and output pins either to rise above Vcc (Vdd) or to fall below GND (Vss). Also, follow any prescribed power-on sequence, so that power is applied gradually or in steps rather than abruptly.

(2) Do not allow any abnormal noise signals to be applied to the device.

(3) Set the voltage levels of unused input pins to Vcc (Vdd) or GND (Vss).

(4) Do not connect output pins to one another.

3.3.6 Input/Output protection

Wired-AND configurations, in which outputs are connected together, cannot be used, since this short-circuits the outputs. Outputs should, of course, never be connected to Vcc (Vdd) or GND (Vss).

Furthermore, ICs with tri-state outputs can undergo performance degradation if a shorted output current is allowed to flow for an extended period of time. Therefore, when designing circuits, make sure that tri-state outputs will not be enabled simultaneously.

3.3.7 Load capacitance

Some devices display increased delay times if the load capacitance is large. Also, large charging and discharging currents will flow in the device, causing noise. Furthermore, since outputs are shorted for a relatively long time, wiring can become fused.

Consult the technical information for the device being used to determine the recommended load capacitance.

3.3.8 Thermal design

The failure rate of semiconductor devices is greatly increased as operating temperatures increase. As shown in Figure 2, the internal thermal stress on a device is the sum of the ambient temperature and the temperature rise due to power dissipation in the device. Therefore, to achieve optimum reliability, observe the following precautions concerning thermal design:

(1) Keep the ambient temperature (Ta) as low as possible.

(2) If the device's dynamic power dissipation is relatively large, select the most appropriate circuit board material, and consider the use of heat sinks or of forced air cooling. Such measures will help lower the thermal resistance of the package.

(3) Derate the device's absolute maximum ratings to minimize thermal stress from power dissipation.
$\theta ja = \theta jc + \theta ca$
$\theta ja = (Tj - Ta) / P$
$\theta jc = (Tj - Tc) / P$
$\theta ca = (Tc - Ta) / P$
in which θja = thermal resistance between junction and surrounding air (°C/W)
θjc = thermal resistance between junction and package surface, or internal thermal resistance (°C/W)
θca = thermal resistance between package surface and surrounding air, or external thermal resistance (°C/W)
Tj = junction temperature or chip temperature (°C)
Tc = package surface temperature or case temperature (°C)
Ta = ambient temperature (°C)
P = power dissipation (W)

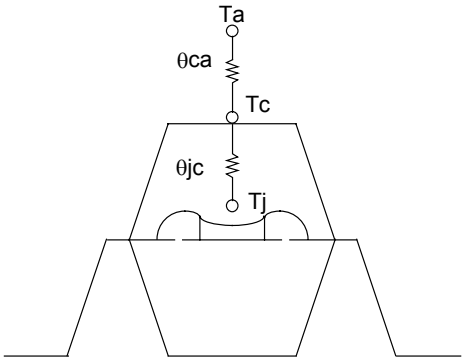


Figure 2 Thermal resistance of package

3.3.9 Interfacing

When connecting inputs and outputs between devices, make sure input voltage (V_{IL}/V_{IH}) and output voltage (V_{OL}/V_{OH}) levels are matched. Otherwise, the devices may malfunction. When connecting devices operating at different supply voltages, such as in a dual-power-supply system, be aware that erroneous power-on and power-off sequences can result in device breakdown. For details of how to interface particular devices, consult the relevant technical datasheets and databooks. If you have any questions or doubts about interfacing, contact your nearest Toshiba office or distributor.

3.3.10 Decoupling

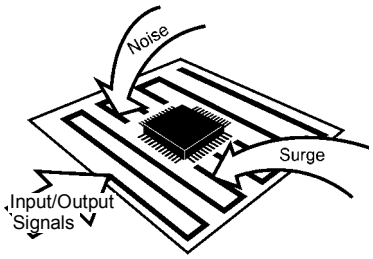
Spike currents generated during switching can cause Vcc (Vdd) and GND (Vss) voltage levels to fluctuate, causing ringing in the output waveform or a delay in response speed. (The power supply and GND wiring impedance is normally 50 Ω to 100 Ω.) For this reason, the impedance of power supply lines with respect to high frequencies must be kept low. This can be accomplished by using thick and short wiring for the Vcc (Vdd) and GND (Vss) lines and by installing decoupling capacitors (of approximately 0.01 μF to 1 μF capacitance) as high-frequency filters between Vcc (Vdd) and GND (Vss) at strategic locations on the printed circuit board.

For low-frequency filtering, it is a good idea to install a 10- to 100-μF capacitor on the printed circuit board (one capacitor will suffice). If the capacitance is excessively large, however, (e.g. several thousand μF) latch-up can be a problem. Be sure to choose an appropriate capacitance value.

An important point about wiring is that, in the case of high-speed logic ICs, noise is caused mainly by reflection and crosstalk, or by the power supply impedance. Reflections cause increased signal delay, ringing, overshoot and undershoot, thereby reducing the device's safety margins with respect to noise. To prevent reflections, reduce the wiring length by increasing the device mounting density so as to lower the inductance (L) and capacitance (C) in the wiring. Extreme care must be taken, however, when taking this corrective measure, since it tends to cause crosstalk between the wires. In practice, there must be a trade-off between these two factors.

3.3.11 External noise

Printed circuit boards with long I/O or signal pattern lines are vulnerable to induced noise or surges from outside sources. Consequently, malfunctions or breakdowns can result from overcurrent or overvoltage, depending on the types of device used. To protect against noise, lower the impedance of the pattern line or insert a noise-canceling circuit. Protective measures must also be taken against surges.



For details of the appropriate protective measures for a particular device, consult the relevant databook.

3.3.12 Electromagnetic interference

Widespread use of electrical and electronic equipment in recent years has brought with it radio and TV reception problems due to electromagnetic interference. To use the radio spectrum effectively and to maintain radio communications quality, each country has formulated regulations limiting the amount of electromagnetic interference which can be generated by individual products.

Electromagnetic interference includes conduction noise propagated through power supply and telephone lines, and noise from direct electromagnetic waves radiated by equipment. Different measurement methods and corrective measures are used to assess and counteract each specific type of noise.

Difficulties in controlling electromagnetic interference derive from the fact that there is no method available which allows designers to calculate, at the design stage, the strength of the electromagnetic waves which will emanate from each component in a piece of equipment. For this reason, it is only after the prototype equipment has been completed that the designer can take measurements using a dedicated instrument to determine the strength of electromagnetic interference waves. Yet it is possible during system design to incorporate some measures for the prevention of electromagnetic interference, which can facilitate taking corrective measures once the design has been completed. These include installing shields and noise filters, and increasing the thickness of the power supply wiring patterns on the printed circuit board. One effective method, for example, is to devise several shielding options during design, and then select the most suitable shielding method based on the results of measurements taken after the prototype has been completed.

3.3.13 Peripheral circuits

In most cases semiconductor devices are used with peripheral circuits and components. The input and output signal voltages and currents in these circuits must be chosen to match the semiconductor device's specifications. The following factors must be taken into account.

(1) Inappropriate voltages or currents applied to a device's input pins may cause it to operate erratically. Some devices contain pull-up or pull-down resistors. When designing your system, remember to take the effect of this on the voltage and current levels into account.

(2) The output pins on a device have a predetermined external circuit drive capability. If this drive capability is greater than that required, either incorporate a compensating circuit into your design or carefully select suitable components for use in external circuits.

3.3.14 Safety standards

Each country has safety standards which must be observed. These safety standards include requirements for quality assurance systems and design of device insulation. Such requirements must be fully taken into account to ensure that your design conforms to the applicable safety standards.

3.3.15 Other precautions

(1) When designing a system, be sure to incorporate fail-safe and other appropriate measures according to the intended purpose of your system. Also, be sure to debug your system under actual board-mounted conditions.

(2) If a plastic-package device is placed in a strong electric field, surface leakage may occur due to the charge-up phenomenon, resulting in device malfunction. In such cases take appropriate measures to prevent this problem, for example by protecting the package surface with a conductive shield.

(3) With some microcomputers and MOS memory devices, caution is required when powering on or resetting the device. To ensure that your design does not violate device specifications, consult the relevant databook for each constituent device.

(4) Ensure that no conductive material or object (such as a metal pin) can drop onto and short the leads of a device mounted on a printed circuit board.

3.4 Inspection, Testing and Evaluation

3.4.1 Grounding

⚠CAUTION Ground all measuring instruments, jigs, tools and soldering irons to earth.
Electrical leakage may cause a device to break down or may result in electric shock.

3.4.2 Inspection Sequence

⚠CAUTION

① Do not insert devices in the wrong orientation. Make sure that the positive and negative electrodes of the power supply are correctly connected. Otherwise, the rated maximum current or maximum power dissipation may be exceeded and the device may break down or undergo performance degradation, causing it to catch fire or explode, resulting in injury to the user.

② When conducting any kind of evaluation, inspection or testing using AC power with a peak voltage of 42.4 V or DC power exceeding 60 V, be sure to connect the electrodes or probes of the testing equipment to the device under test before powering it on. Connecting the electrodes or probes of testing equipment to a device while it is powered on may result in electric shock, causing injury.

(1) Apply voltage to the test jig only after inserting the device securely into it. When applying or removing power, observe the relevant precautions, if any.

(2) Make sure that the voltage applied to the device is off before removing the device from the test jig. Otherwise, the device may undergo performance degradation or be destroyed.

(3) Make sure that no surge voltages from the measuring equipment are applied to the device.

(4) The chips housed in tape carrier packages (TCPs) are bare chips and are therefore exposed. During inspection take care not to crack the chip or cause any flaws in it.
Electrical contact may also cause a chip to become faulty. Therefore make sure that nothing comes into electrical contact with the chip.

3.5 Mounting

There are essentially two main types of semiconductor device package: lead insertion and surface mount. During mounting on printed circuit boards, devices can become contaminated by flux or damaged by thermal stress from the soldering process. With surface-mount devices in particular, the most significant problem is thermal stress from solder reflow, when the entire package is subjected to heat. This section describes a recommended temperature profile for each mounting method, as well as general precautions which you should take when mounting devices on printed circuit boards. Note, however, that even for devices with the same package type, the appropriate mounting method varies according to the size of the chip and the size and shape of the lead frame. Therefore, please consult the relevant technical datasheet and databook.

3.5.1 Lead forming

⚠CAUTION

① Always wear protective glasses when cutting the leads of a device with clippers or a similar tool. If you do not, small bits of metal flying off the cut ends may damage your eyes.

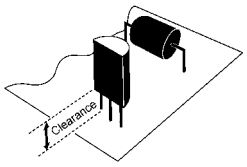
② Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger.

Semiconductor devices must undergo a process in which the leads are cut and formed before the devices can be mounted on a printed circuit board. If undue stress is applied to the interior of a device during this process, mechanical breakdown or performance degradation can result. This is attributable primarily to differences between the stress on the device's external leads and the stress on the internal leads. If the relative difference is great enough, the device's internal leads, adhesive properties or sealant can be damaged. Observe these precautions during the lead-forming process (this does not apply to surface-mount devices):

(1) Lead insertion hole intervals on the printed circuit board should match the lead pitch of the device precisely.

(2) If lead insertion hole intervals on the printed circuit board do not precisely match the lead pitch of the device, do not attempt to forcibly insert devices by pressing on them or by pulling on their leads.

(3) For the minimum clearance specification between a device and a printed circuit board, refer to the relevant device's datasheet and databook. If necessary, achieve the required clearance by forming the device's leads appropriately. Do not use the spacers which are used to raise devices above the surface of the printed circuit board during soldering to achieve clearance. These spacers normally continue to expand due to heat, even after the solder has begun to solidify; this applies severe stress to the device.



(4) Observe the following precautions when forming the leads of a device prior to mounting.

- Use a tool or jig to secure the lead at its base (where the lead meets the device package) while bending so as to avoid mechanical stress to the device. Also avoid bending or stretching device leads repeatedly.
- Be careful not to damage the lead during lead forming.
- Follow any other precautions described in the individual datasheets and databooks for each device and package type.

3.5.2 Socket mounting

(1) When socket mounting devices on a printed circuit board, use sockets which match the inserted device's package.

(2) Use sockets whose contacts have the appropriate contact pressure. If the contact pressure is insufficient, the socket may not make a perfect contact when the device is repeatedly inserted and removed; if the pressure is excessively high, the device leads may be bent or damaged when they are inserted into or removed from the socket.

(3) When soldering sockets to the printed circuit board, use sockets whose construction prevents flux from penetrating into the contacts or which allows flux to be completely cleaned off.

(4) Make sure the coating agent applied to the printed circuit board for moisture-proofing purposes does not stick to the socket contacts.

(5) If the device leads are severely bent by a socket as it is inserted or removed and you wish to repair the leads so as to continue using the device, make sure that this lead correction is only performed once. Do not use devices whose leads have been corrected more than once.

(6) If the printed circuit board with the devices mounted on it will be subjected to vibration from external sources, use sockets which have a strong contact pressure so as to prevent the sockets and devices from vibrating relative to one another.

3.5.3 Soldering temperature profile

The soldering temperature and heating time vary from device to device. Therefore, when specifying the mounting conditions, refer to the individual datasheets and databooks for the devices used.

(1) Using medium infrared ray reflow

- Heating top and bottom with long or medium infrared rays is recommended (see Figure 3).

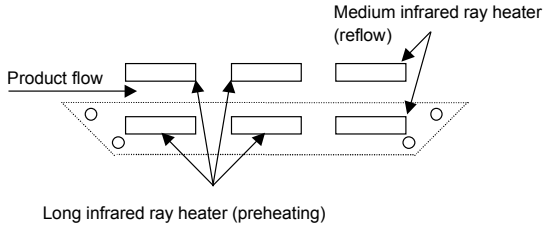


Figure 3 Heating top and bottom with long or medium infrared rays

- Complete the infrared ray reflow process for 30 to 50 seconds at a package surface temperature of between 230°C and 260°C.
- Refer to Figure 4 for an example of a good temperature profile for infrared or hot air reflow.

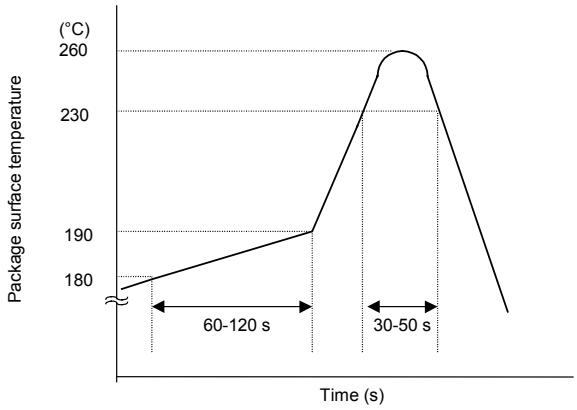


Figure 4 Sample temperature profile (Pb free) for infrared or hot air reflow

(2) Using hot air reflow

- Complete hot air reflow for 30 to 50 seconds at a package surface temperature of between 230°C and 260°C.
- For an example of a recommended temperature profile, refer to Figure 4 above.

3.5.4 Flux cleaning and ultrasonic cleaning

(1) When cleaning circuit boards to remove flux, make sure that no residual reactive ions such as Na or Cl remain. Note that organic solvents react with water to generate hydrogen chloride and other corrosive gases which can degrade device performance.

(2) Washing devices with water will not cause any problems. However, make sure that no reactive ions such as sodium and chlorine are left as a residue. Also, be sure to dry devices sufficiently after washing.

(3) Do not rub device markings with a brush or with your hand during cleaning or while the devices are still wet from the cleaning agent. Doing so can rub off the markings.

(4) The dip cleaning, shower cleaning and steam cleaning processes all involve the chemical action of a solvent. Use only recommended solvents for these cleaning methods. When immersing devices in a solvent or steam bath, make sure that the temperature of the liquid is 50°C or below, and that the circuit board is removed from the bath within one minute.

(5) Ultrasonic cleaning should not be used with hermetically-sealed ceramic packages such as a leadless chip carrier (LCC), pin grid array (PGA) or charge-coupled device (CCD), because the bonding wires can become disconnected due to resonance during the cleaning process. Even if a device package allows ultrasonic cleaning, limit the duration of ultrasonic cleaning to as short a time as possible, since long hours of ultrasonic cleaning degrade the adhesion between the mold resin and the frame material. The following ultrasonic cleaning conditions are recommended:

Frequency: 27 kHz ~ 29 kHz

Ultrasonic output power: 300 W or less (0.25 W/cm^2 or less)

Cleaning time: 30 seconds or less

Suspend the circuit board in the solvent bath during ultrasonic cleaning in such a way that the ultrasonic vibrator does not come into direct contact with the circuit board or the device.

3.5.5 No cleaning

If analog devices or high-speed devices are used without being cleaned, flux residues may cause minute amounts of leakage between pins. Similarly, dew condensation, which occurs in environments containing residual chlorine when power to the device is on, may cause between-lead leakage or migration. Therefore, Toshiba recommends that these devices be cleaned.
However, if the flux used contains only a small amount of halogen (0.05W% or less), the devices may be used without cleaning without any problems. No cleaning is recommended for TX4938.

3.5.6 Mounting tape carrier packages (TCPs)

(1) When tape carrier packages (TCPs) are mounted, measures must be taken to prevent electrostatic breakdown of the devices.

(2) If devices are being picked up from tape, or outer lead bonding (OLB) mounting is being carried out, consult the manufacturer of the insertion machine which is being used, in order to establish the optimum mounting conditions in advance and to avoid any possible hazards.

(3) The base film, which is made of polyimide, is hard and thin. Be careful not to cut or scratch your hands or any objects while handling the tape.

(4) When punching tape, try not to scatter broken pieces of tape too much.

(5) Treat the extra film, reels and spacers left after punching as industrial waste, taking care not to destroy or pollute the environment.

(6) Chips housed in tape carrier packages (TCPs) are bare chips and therefore have their reverse side exposed. To ensure that the chip will not be cracked during mounting, ensure that no mechanical shock is applied to the reverse side of the chip. Electrical contact may also cause a chip to fail. Therefore, when mounting devices, make sure that nothing comes into electrical contact with the reverse side of the chip.
If your design requires connecting the reverse side of the chip to the circuit board, please consult Toshiba or a Toshiba distributor beforehand.

3.5.7 Mounting chips

Devices delivered in chip form tend to degrade or break under external forces much more easily than plastic-packaged devices. Therefore, caution is required when handling this type of device.

(1) Mount devices in a properly prepared environment so that chip surfaces will not be exposed to polluted ambient air or other polluted substances.

(2) When handling chips, be careful not to expose them to static electricity.
In particular, measures must be taken to prevent static damage during the mounting of chips. With this in mind, Toshiba recommend mounting all peripheral parts first and then mounting chips last (after all other components have been mounted).

(3) Make sure that PCBs (or any other kind of circuit board) on which chips are being mounted do not have any chemical residues on them (such as the chemicals which were used for etching the PCBs).

(4) When mounting chips on a board, use the method of assembly that is most suitable for maintaining the appropriate electrical, thermal and mechanical properties of the semiconductor devices used.

* For details of devices in chip form, refer to the relevant device's individual datasheets.

3.5.8 Circuit board coating

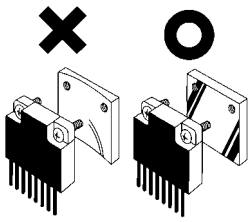
When devices are to be used in equipment requiring a high degree of reliability or in extreme environments (where moisture, corrosive gas or dust is present), circuit boards may be coated for protection. However, before doing so, you must carefully consider the possible stress and contamination effects that may result and then choose the coating resin which results in the minimum level of stress to the device.

3.5.9 Heat sinks

(1) When attaching a heat sink to a device, be careful not to apply excessive force to the device in the process.

(2) When attaching a device to a heat sink by fixing it at two or more locations, evenly tighten all the screws in stages (i.e. do not fully tighten one screw while the rest are still only loosely tightened). Finally, fully tighten all the screws up to the specified torque.

(3) Drill holes for screws in the heat sink exactly as specified. Smooth the surface by removing burrs and protrusions or indentations which might interfere with the installation of any part of the device.



(4) A coating of silicone compound can be applied between the heat sink and the device to improve heat conductivity. Be sure to apply the coating thinly and evenly; do not use too much. Also, be sure to use a non-volatile compound, as volatile compounds can crack after a time, causing the heat radiation properties of the heat sink to deteriorate.

(5) If the device is housed in a plastic package, use caution when selecting the type of silicone compound to be applied between the heat sink and the device. With some types, the base oil separates and penetrates the plastic package, significantly reducing the useful life of the device.
Two recommended silicone compounds in which base oil separation is not a problem are YG6260 from Toshiba Silicone.

(6) Heat-sink-equipped devices can become very hot during operation. Do not touch them, or you may sustain a burn.

3.5.10 Tightening torque

(1) Make sure the screws are tightened with fastening torques not exceeding the torque values stipulated in individual datasheets and databooks for the devices used.

(2) Do not allow a power screwdriver (electrical or air-driven) to touch devices.

3.5.11 Repeated device mounting and usage

Do not remount or re-use devices which fall into the categories listed below; these devices may cause significant problems relating to performance and reliability.

(1) Devices which have been removed from the board after soldering

(2) Devices which have been inserted in the wrong orientation or which have had reverse current applied

(3) Devices which have undergone lead forming more than once

3.6 Protecting Devices in the Field

3.6.1 Temperature

Semiconductor devices are generally more sensitive to temperature than are other electronic components. The various electrical characteristics of a semiconductor device are dependent on the ambient temperature at which the device is used. It is therefore necessary to understand the temperature characteristics of a device and to incorporate device derating into circuit design. Note also that if a device is used above its maximum temperature rating, device deterioration is more rapid and it will reach the end of its usable life sooner than expected.

3.6.2 Humidity

Resin-molded devices are sometimes improperly sealed. When these devices are used for an extended period of time in a high-humidity environment, moisture can penetrate into the device and cause chip degradation or malfunction. Furthermore, when devices are mounted on a regular printed circuit board, the impedance between wiring components can decrease under high-humidity conditions. In systems which require a high signal-source impedance, circuit board leakage or leakage between device lead pins can cause malfunctions. The application of a moisture-proof treatment to the device surface should be considered in this case. On the other hand, operation under low-humidity conditions can damage a device due to the occurrence of electrostatic discharge. Unless damp-proofing measures have been specifically taken, use devices only in environments with appropriate ambient moisture levels (i.e. within a relative humidity range of 40% to 60%).

3.6.3 Corrosive gases

Corrosive gases can cause chemical reactions in devices, degrading device characteristics.
For example, sulphur-bearing corrosive gases emanating from rubber placed near a device (accompanied by condensation under high-humidity conditions) can corrode a device's leads. The resulting chemical reaction between leads forms foreign particles which can cause electrical leakage.

3.6.4 Radioactive and cosmic rays

Most industrial and consumer semiconductor devices are not designed with protection against radioactive and cosmic rays. Devices used in aerospace equipment or in radioactive environments must therefore be shielded.

3.6.5 Strong electrical and magnetic fields

Devices exposed to strong magnetic fields can undergo a polarization phenomenon in their plastic material, or within the chip, which gives rise to abnormal symptoms such as impedance changes or increased leakage current. Failures have been reported in LSIs mounted near malfunctioning deflection yokes in TV sets. In such cases the device's installation location must be changed or the device must be shielded against the electrical or magnetic field. Shielding against magnetism is especially necessary for devices used in an alternating magnetic field because of the electromotive forces generated in this type of environment.

3.6.6 Interference from light (ultraviolet rays, sunlight, fluorescent lamps and incandescent lamps)

Light striking a semiconductor device generates electromotive force due to photoelectric effects. In some cases the device can malfunction. This is especially true for devices in which the internal chip is exposed. When designing circuits, make sure that devices are protected against incident light from external sources. This problem is not limited to optical semiconductors and EPROMs. All types of device can be affected by light.

3.6.7 Dust and oil

Just like corrosive gases, dust and oil can cause chemical reactions in devices, which will adversely affect a device's electrical characteristics. To avoid this problem, do not use devices in dusty or oily environments. This is especially important for optical devices because dust and oil can affect a device's optical characteristics as well as its physical integrity and the electrical performance factors mentioned above.

3.6.8 Fire

Semiconductor devices are combustible; they can emit smoke and catch fire if heated sufficiently. When this happens, some devices may generate poisonous gases. Devices should therefore never be used in close proximity to an open flame or a heat-generating body, or near flammable or combustible materials.

3.7 Disposal of Devices and Packing Materials

When discarding unused devices and packing materials, follow all procedures specified by local regulations in order to protect the environment against contamination.

4. Precautions and Usage Considerations

This section describes matters specific to each product group which need to be taken into consideration when using devices. If the same item is described in Sections 3 and 4, the description in Section 4 takes precedence.

4.1 Microcontrollers

4.1.1 Design

(1) Using resonators which are not specifically recommended for use

Resonators recommended for use with Toshiba products in microcontroller oscillator applications are listed in Toshiba databooks along with information about oscillation conditions. If you use a resonator not included in this list, please consult Toshiba or the resonator manufacturer concerning the suitability of the device for your application.

(2) Undefined functions

In some microcontrollers certain instruction code values do not constitute valid processor instructions. Also, it is possible that the values of bits in registers will become undefined. Take care in your applications not to use invalid instructions or to let register bit values become undefined.

TMPR4938

2005-3 Rev 2.0

Conventions in this Manual

Value Conventions

- Hexadecimal values are expressed as in the following example. (This value is expressed as 42 in the decimal system.)
- KB (kilobyte) = 1,024 Bytes,
  MB (megabyte) = 1,024 × 1,024 = 1,048,576 Bytes,
  GB (gigabyte) = 1,024 × 1,024 × 1,024 = 1,073,741,824 Bytes

Data Conventions

- Byte: 8 bits
- Half-word: 2 consecutive Bytes (16 bits)
- Word: 4 consecutive Bytes (32 bits)
- Double-word: 8 consecutive Bytes (64 bits)

Signal Conventions

- An asterisk ("*") is added to the end of signal names to indicate Low Active signals. (Example: RESET*)
- "Assert" means to move a signal to its Active level. "Deassert" means to move a signal to its Inactive level.

Register Conventions

- Bit operation is expressed as follows.
  Set: Put a bit in the "1" position.
  Clear: Put a bit in the "0" position.
- Properties of each bit in a register are expressed as follows.
  R: Read only. The software cannot change the bit value.
  W: Write only. The value that is read is undefined.
  R/W: Read/Write is possible.
  W1C Write 1 Clear. This corresponding bit is cleared when "1" is written to this bit. "0" is invalid if written.
  R/W1C: Read/Write 1 Clear. These bits can be read from and written to. The corresponding bit is cleared when "1" is written to this bit. "0" is invalid if written.
  R/W0C: Read/Write 0 Clear. These bits can be read from and written to. The corresponding bit is cleared when "0" is written to this bit. "1" is invalid if written.
  R/W1S Read/Write 1 Set. These bits can be read from and written to. The corresponding bit is set when "1" is written to this bit. "0" is invalid if written.
  RS/WC Read Set/Write Clear. These bits can be read from and written to. The bits is set when read, and a write of an arbitrary value to the bit clears it.
  R/L: Property unique to the PCI Controller. This bit can be read. The value of this bit can only be changed by the method described in "10.3.14: Set Configuration Space".
- Registers and the register bit/field name are expressed as "*<register name>.<bit/field name>*".
  Example: CCFG.TOE
  The above example indicates Time Out Bus Error Enable (TOE), a bit field of bit 14 in the Chip Configuration Register (CCFG).

Handling reserved regions

Operation is undefined when a register defined in this document as a reserved region (Reserved) is accessed. If there is a bit or field that was defined as Reserved in a register, write the expressed default value or the specified value ("0" if no particular value is expressed) to these bits. Also, do not use any value read from this bit/field.

Diagnostic function

Any function described as a "diagnostic function" is used to facilitate operation evaluations. The operation of such functions is not guaranteed.

References

64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture
(http://www.semicon.toshiba.co.jp/eng/index.html)

MIPS RISC Architecture, Gerry Kane and Joe Heinrich (ISBN 0-13-590472-2)

See MIPS Run, Dominic Sweetman (ISBN 1-55860-410-3)

MIPS Publications (http://www.mips.com/publications/)

PCI Local Bus Specification Revision 2.2 (http://www.pcisig.com/)

PCI Bus Power Management Interface Specification Revision 1.1

Audio CODEC '97 (AC '97) Revision 2.1 (http://developer.intel.com/ial/scalableplatforms/audio/)

– SmartMedia™ Physical Format Specifications Web-Online Version 1.00

– SMIL (SmartMedia™ Interface Library) Software Edition Version 1.00

(http://www.ssfdc.or.jp/english/business/)

1. Overview and Features

1.1 Overview

The TMPR4938 (TX4938) is a standard microcontroller that belongs to the 64-bit TX System RISC TX49 family.

The TX4938 uses the TX49/H3 core as its CPU. The TX49/H3 core is a 64-bit RISC core that Toshiba developed based on the MIPS III architecture of MIPS Technologies, Inc. (MIPS). For details of the TX49/H3 core such as instruction sets, see "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture".

In addition to the TX49/H3 core, the TX4938 uses as microcontrollers for embedded applications an External Bus Controller, DMA Controllers, SDRAM Controller, PCI Controller, Ethernet Controller and NAND Flash Memory Controller (enables booting from NAND flash memory). The TX4938 also has peripheral circuits such as a synchronous serial interface, on-chip SRAM, a serial I/O port, a timer/counter, a parallel I/O port, an AC-link Controller and an Interrupt Controller. By having an on-chip a data bus with a maximum width of 64 bits and an SDRAM Controller with a memory clock frequency of 133 MHz, the TX4938 realizes low memory access latency and high memory bandwidth. This allows the TX4938 to show its capacity as a high-performance CPU core.

1.2 Features

- TX49/H3 core
  Maximum Operating Frequency: 300 MHz (TMPR4938XBG-300), 333 MHz (TMPR4938XBG-333)
  On-chip IEEE754-compliant single/double precision floating point unit (FPU) function
- External Bus Controller (8 channels)
- Supports the ISA/ATA interface
- DMA (Direct Memory Access) Controllers (8 channels)
- SDRAM Controller (4 channels)
  64-bit Data Bus
  Memory Clock Frequency: 133 MHz (For relationship between CPU clock and memory clock, see Section 6.1)
  Supports ECC/Parity
- PCI Controller
  Complies with PCI Local Bus Specification Revision 2.2
  PCI Bus Clock Frequency: 66 MHz/33 MHz
- Ethernet Controller (2 channels)
- NAND Flash Memory Controller (bootable)
- Serial Peripheral Interface (SPI)
- On-chip SRAM (2 K-byte)
- Serial I/O Port (2 channels)
- Timer/Counter (3 channels)
- Parallel I/O Port (Maximum 16 bits)
- AC-link Controller
- Interrupt Controller
- Selectable Little Endian mode or Big Endian mode

- Low Power Consumption
  Supports internal 1.5 V, IO block 3.3 V operation, and low power consumption mode (Halt)
- Supports IEEE1149.1 (JTAG): Debugging Support Unit (EJTAG)
- Package: 484-pin PBGA (with 64pin thermal ball)

1.2.1 Features of the TX49/H3 core

The TX49/H3 core is a high-performance, low power consumption 64-bit RISC CPU core that Toshiba developed.

- 64-bit operation
- 32 64-bit integer general-purpose registers
- 64 GB physical address space
- Optimized 5-stage pipeline
- Instruction Set
  Upwards compatible MIPS III ISA
  Added 3-operand multiply instruction, MAC (multiply accumulate) instruction, and PREF (pre-fetch) instruction
- Supports 32 KB Instruction cache, 32 KB Data cache, 4-way set associative, and the lock function
- MMU (Memory Management Unit)
  48 double entry (odd/even) joint TLB
- On-chip IEEE754-compatible single-precision and double-precision FPU
- 4-stage write buffer mounted
- Debugging Support Unit: EJTAG

1.2.2 Features of TX4938 peripheral functions

(1) External Bus Controller (EBUSC)

The External Bus Controller generates the signals necessary to control external memory and external I/O devices.

- Has 8-channel Chip Select signal, can control up to 8 external devices
- Supports access of ROM (mask ROM, page mode ROM, EPROM, EEPROM), SRAM, Flash ROM, and I/O devices
- Can set data bus width to 32 bits, 16 bits, or 8 bits for each channel
- System clock for External Bus Controller (SYSCLK) frequency is up to 133 MHz (For relationship between CPU clock and this system clock, see Section 6.1)
- Can select full speed, 1/2 speed, 1/3 speed , or 1/4 speed for each channel
- Can set timing for each channel
  Can set up and set the Hold interval of the Address, Chip Enable, Write Enable, and Output Enable signals
- Supports access of devices with a 32-bit wide data bus in memory sizes from 1 MB to 1 GB. Supports access of devices with a 16-bit wide data bus in memory sizes from 1 MB to 512 MB. Supports access of devices with an 8-bit wide data bus in memory sizes from 1 MB to 256 MB.
- Supports the ISA/ATA interface

(2) Direct Memory Access Controllers (DMAC)

The TX4938 has two DMA Controllers for invoking DMA transfer with memory and I/O devices. Each DMA Controller has 4 built-in DMA Channels.

- Can set internal/external DMA requests
- Supports as internal DMA requests DMA with the on-chip Serial I/O Controller or AC-link Controller
- Supports as external I/O DMA transfer modes using external DMA requests Single Address transfer (Fly-by DMA) and Dual Address transfer
- Supports transfer between external devices with a data bus width of 32 bits, 16 bits or 8 bits and memory
- Supports memory-memory copy mode that has no address boundary constraints Can perform Burst transfer of up to 8 double words in a single read or write operation
- Supports the Memory Fill mode that writes double-word data to the memory region
- Supports Chain DMA transfer

(3) SDRAM Controller (SDRAMC)

The SDRAM Controller generates the control signals required for the SDRAM interface. By having 4 on-chip channels and supporting a variety of memory configurations, the SDRAM Controller can support memory sizes of up to 4 GB (1 GB/channel).

- Memory clock (SDCLK) frequencies from 50 MHz to 133 MHz (For relationship between CPU clock and memory clock, see Section 6.1)
- 4 sets of independent memory channels
- Supports 2-bank or 4-bank 16 MB, 64 MB, 128 MB, 256 MB, or 512 MB SDRAM
- Can used Registered DIMM
- Supports ECC or parity generation/check functions
- Can select either 32-bit or 64-bit data bus width for each channel
- Can set SDRAM timing for each channel
- Supports TX49/H3 core critical word first access
- Low power consumption mode: can select Self-refresh or Pre-charge power down

(4) PCI Controller (PCIC)

The TX4938 has an on-chip PCI Controller that is compliant with PCI Local Bus Specification Revision 2.2.

- PCI Local Bus Specification Revision 2.2 compliant
- 32-bit PCI interface with maximum PCI Bus clock frequency of 66 MHz
- Supports both the Target and Initiator functions
- Can change the address mapping between the internal bus and PCI Bus
- Has an on-chip PCI Bus arbiter and can connect up to 4 External Bus Masters
- Has a function mounted for booting the TX4938 from memory on the PCI Bus
- Has an on-chip 1-channel PCI Controller-dedicated DMA Controller (PDMAC)

(5) Serial I/O port (SIO)

The TX4938 has an on-chip 2-channel asynchronous serial I/O interface (full duplex UART)

- Full duplex UART × 2 channels
- On-chip baud rate generator
- FIFO
  Transmission: on-chip 8-bit × 8-stage FIFO
  Reception: on-chip 13-bit × 16-stage (data: 8 bits; status: 5 bits) FIFO
- Supports DMA transfer

(6) Timer/Counter control (TMR)

The TX4938 has an on-chip 3-channel timer/counter

- 32-bit setup counter × 3 channels
- Supports 3 modes: Interval Timer mode, Pulse Generator mode, Watchdog Timer mode
- Timer output pins: 2 pins
- Count clock input pin: 1 pin
- Watchdog external reset pin: 1 pin

(7) Parallel I/O port (PIO)

The TX4938 has a 16-bit parallel I/O port (8 bits of which are shared with CB[7:0])

- Can set I/O direction and port type (totem pole output/open drain output) during output for each bit

(8) AC-link Controller (ACLC)

The TX4938 on-chip AC-link Controller can connect and manipulate audio and/or modem CODECs described in “Audio CODEC ’97 Revision 2.1”.

- Supports up to 2 CODECs
- Supports 16-bit PCM stereo channel recording and playback
- Supports 16-bit surround, center, LFE channel playback
- Supports variable rate audio recording and playback
- Supports Line 1 for modem CODEC and GPIO slot
- Supports AC-link low-power mode, Wake Up, and Warm Reset

Supports sample data input/output by DMA transfer

(9) Interrupt Controller (IRC)

The Interrupt Controller built into the TX4938 receives interrupt requests and external interrupts from the TX4938 on-chip peripheral circuits and issues interrupt requests to the TX49/H3 core. This controller has a 16-bit flag register that generates interrupt requests to an external device or the TX49/H3 core.

- Supports 19 types of internal interrupts from the on-chip peripheral circuits and 6 external interrupt signal inputs
- Sets 8 priority levels for each interrupt input

- Can select either the Edge or Level interrupt detection mode for each external interrupt
- Has a built-in 16-bit read/write register as a flag register for interrupt requests. Can request interrupts to an external device or to the TX49/H3 core (IRC interrupt)

(10) Ethernet Controller

- 2-channel on-chip Ethernet interface
- Supports transfer rates of 100 Mbps or 10 Mbps
- Supports MAC layers, employs the standard interface MII between physical layers and MAC layers
- On-chip DMA

(11) NAND Flash Memory Controller

- Controls the NAND Flash Memory Interface by the register settings
- On-chip ECC (Error Correction Code) calculation circuit
- On-chip mask ROM, is bootable from NAND Flash Memory

(12) Serial Peripheral Interface (SPI)

- Full duplex, synchronous serial data transfer (I/O data, clock signals)
- Can specify 8-bit or 16-bit data length
- Programmable SPI baud rate
- Supports only Masters

(13) On-chip SRAM

- On-chip 2 KB SRAM

(14) Extended EJTAGInterface

The TX4938's Extended EJTAG (Extended Enhanced Joint Test Action Group) interface provides two functions: IEEE1149.1-compliant JTAG boundary scan testing and real-time debugging using the Debugging Support Unit (DSU) built into the TX49/H3 core.

- IEEE 1149.1 JTAG Boundary Scan
- Can use execution control (invoke, break, step, register/memory access) or PC tracing as real-time debugging function that uses a dedicated emulation probe

2. Configuration

2.1 TX4938 block diagram

Figure 2.1.1 is an internal block diagram of the TX4938.

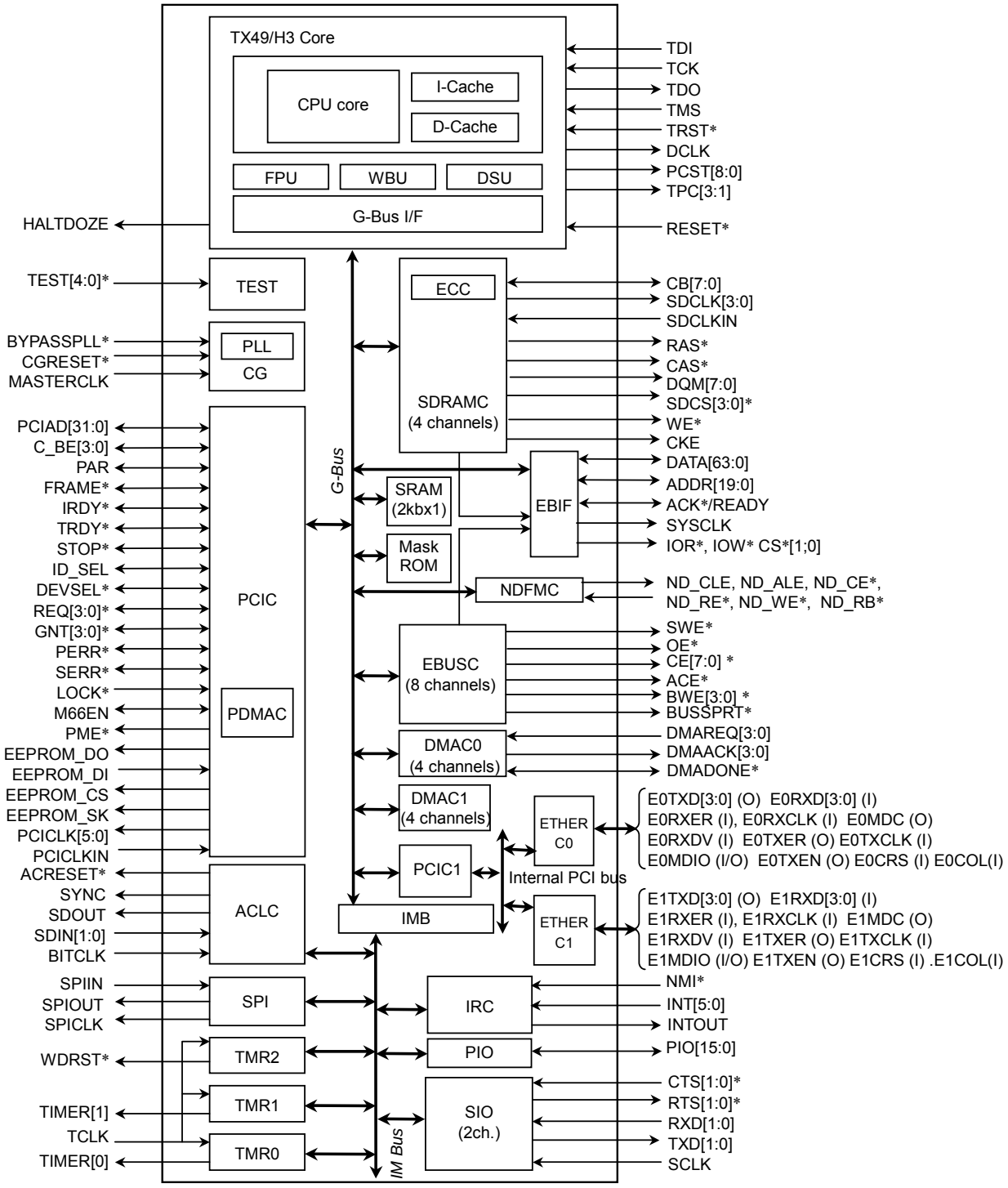


Figure 2.1.1 TX4938 Block Diagram

The TX4938 has the following blocks.

(1) TX49/H3 Core: Consists of a CPU, System Control Coprocessor (CP0), Instruction cache, Data cache, Floating-point Unit (FPU), write buffer (WBU), Debugging Support Unit (DSU), and a G-Bus I/F.

- FPU: An IEEE754-compliant single-precision or double-precision floating-point unit. Is allocated as one coprocessor unit (CP1).
- I-Cache: Instruction cache memory. 32 KB, 4-way set associative.
- D-Cache: Data cache memory. 32 KB, 4-way set associative. You can select as a write policy Write Back, Write Through No Write Allocate, or Write Through Write Allocate.
- DSU: Debugging Support Unit. This is an on-chip debugging module.

(2) EBUSC: External Bus Controller. Controls 8-channel ROM, SRAM, I/O.

(3) DMAC0: Direct Memory Access Controller. Has 4 channels. Can transfer internal I/O device, external I/O device, mutual memory data.

(4) DMAC1: Direct Memory Access Controller. Has 4 channels. Can transfer internal I/O device, mutual memory data.

(5) SDRAMC: SDRAM Controller. Controls 4-channel SDRAM. Supports 64-bit data bus, 133 MHz operation, ECC/parity.

(6) PCIC: PCI Controller. Complies with PCI Local Bus Specification Revision 2.2. Supports 66 MHz operation. Has an on-chip dedicated DMA Controller (PDMAC).

(7) SIO: Serial I/O. This is a 2-channel asynchronous serial I/F.

(8) TMR: Timer/counter. This is a 3-channel timer/counter.

(9) PIO: Parallel I/O. Has an 8-bit dedicated port and an 8-bit shared port.

(10) ACLC: AC-link Controller. Is compliant to Audio CODEC '97 Revision 2.1 (AC'97).

(11) IRC: Interrupt Controller.

(12) EBIF: External Bus Interface. Connects 20-bit External Address Bus or 64-bit External Data Bus to SDRAMC or EBUSC.

(13) CG: Clock Generator. Has a built-in PLL and provides a clock to each TX4938 block.

(14) G-Bus: TX4938 internal bus. This is a 64-bit, high-speed internal bus that is directly connected to the TX49/H3 core.

(15) IM-Bus: TX4938 internal bus. This is a 32-bit, slow-speed internal bus that is connected to the G-Bus via IMB.

(16) MB: G-Bus - IM-Bus Bridge

(17) NDFMC: NAND Flash Memory Controller

(18) Mask ROM: Mask ROM that boots from NAND Flash memory

(19) SRAM: 2 KB on-chip SRAM

(20) ETHERC0: Channel 0 of the Ethernet Controller. The system side is the PCI Bus Interface.

(21) ETHERC1: Channel 1 of the Ethernet Controller. The system side is the PCI Bus Interface.

(22) PCIC1: PCI Bus/G-Bus bridge for connecting ETHERC0 and ETHERC1 to the G-Bus.

(23) SPI: Serial Peripheral Interface. Supports Master operation.

(24) TEST This is the internal diagnostic module.

3. Signals

3.1 Pin Signal Description

In the following tables, asterisks at the end of signal names indicate active-low signals.

In the Type column, PU indicates that the pin is equipped with an internal pull-up resister and PD indicates that the pin is equipped with an internal pull-down resister. OD indicates an open-drain pin.

The Initial State column shows the state of the signal when the RESET* signal is asserted and immediately after it is deasserted. Those signals which are selected by a configuration signal upon a reset have the state selected by the configuration signal even when the reset signal is asserted.

3.1.1 Signals Common to SDRAM and External Bus Interfaces

Table 3.1.1 Signals Common to SDRAM and External Bus Interfaces

Signal Name	Type	Description	Initial State
ADDR[19:0]	Input/output PU	Address Address signals. For SDRAM, ADDR[19:5] are used (refer to Sections “9.3.2.2” and “9.3.2.3 Address Signal Mapping”). When the external bus controller uses these pins, the meaning of each bit varies with the data bus width (refer to Section “7.3.5 Data Bus Size”). The ADDR signals are also used as boot configuration signals (input) during a reset. For details of configuration signals, refer to Section “3.2 Boot Configuration”. The ADDR signals are input signals only when the RESET* signal is asserted and become output signals after the RESET* signal is deasserted.	Input
DATA[63:0]	Input/output PU	Data 64-bit data bus. The DATA[15:0] signals are also used as boot configuration signals (input) during a reset. For details of configuration signals, refer to Section “3.2 Boot Configuration”.	Input
BUSSPRT*	Output	Bus Separate Controls the connection and separation of devices controlled by the external bus controller to or from a high-speed device, such as SDRAM (refer to Section “7.6 Flash ROM, SRAM Usage Example”). H: Separate devices other than SDRAM from the data bus. L: Connect devices other than SDRAM to the data bus. Separation and connection are performed using external bidirectional bus buffers (such as the 74xx245).	High

3.1.2 SDRAM Interface Signals

Table 3.1.2 SDRAM Interface Signals

Signal Name	Type	Description	Initial State
SDCLK[3:0]	Output	SDRAM Controller Clock Clock signals used by SDRAM. The clock frequency is the same as the G-Bus clock (GBUSCLK) frequency. When these clock signals are not used, the pins can be set to H using the SDCLK Enable field of the configuration register (CCFG.SDCLKEN[3:0]).	All High
SDCLKIN	Input/output	SDRAM Feedback Clock input Feedback clock signal for SDRAM controller input signals. Setting the SDCLKINEN bit of the pin configuration register causes the TX4938 to feed back signals internally, making SDCLKIN an output signal.	Input
CKE	Output	Clock Enable CKE signal for SDRAM.	High
SDCS[3:0]*	Output	Synchronous Memory Device Chip Select Chip select signals for SDRAM.	All High
RAS*	Output	Row Address Strobe RAS signal for SDRAM.	High
CAS*	Output	Column Address Strobe CAS signal for SDRAM.	High
WE*	Output	Write Enable WR signal for SDRAM.	High
DQM[7:0]	Output	Data Mask During a write cycle, the DQM signals function as a data mask. During a read cycle, they control the SDRAM output buffers. The bits correspond to the following data bus signals: DQM[7]:DATA[63:54], DQM[6]:DATA[53:48] DQM[5]:DATA[47:40], DQM[4]:DATA[39:32] DQM[3]:DATA[31:24], DQM[2]:DATA[23:16] DQM[1]:DATA[15:8], DQM[0]:DATA[7:0] Connect any one of the DQM[3:0] to SDRAM which connects CB.	All High
CB[7:0]	Input/output PU	ECC/Parity Check Bit ECC/parity check bit signals. The bits correspond to the following data bus signals: CB[7]:DATA[63:54], CB[6]:DATA[53:48] CB[5]:DATA[47:40], CB[4]:DATA[39:32] CB[3]:DATA[31:24], CB[2]:DATA[23:16] CB[1]:DATA[15:8], CB[0]:DATA[7:0] CB[7:0] share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Input

3.1.3 External Interface Signals

Table 3.1.3 External Interface Signals

Signal Name	Type	Description	Initial State
SYSCLK	Output	System Clock Clock for external I/O devices. Outputs a clock in full speed mode (at the same frequency as the G-Bus clock (GBUSCLK) frequency), half speed mode (at one half the GBUSCLK frequency), third speed mode (at one third the GBUSCLK frequency), or quarter speed mode (at one quarter the GBUSCLK frequency). The boot configuration signals on the ADDR[14:13] pins select which speed mode will be used. When this clock signal is not used, the pin can be set to H using the SYSCLK Enable bit of the configuration register (CCFG.SYSCLKEN).	High
ACE*	Output	Address Clock Enable Latch enable signal for the high-order address bits of ADDR.	High
CE[7:0]*	Output	Chip Enable Chip select signals for ROM, SRAM, and I/O devices (refer to Section “3.3 Pin multiplex”).	All High
OE*	Output	Output Enable Output enable signal for ROM, SRAM, and I/O devices.	High
SWE*	Output	Write Enable Write enable signal for SRAM and I/O devices.	High
BWE[3:0]* /BE[3:0]*	Output	Byte Enable/Byte Write Enable BE[3:0]* indicate a valid data position on the data bus DATA[31:0] during read and write bus operation. In 16-bit bus mode, only BE[1:0]* are used. In 8-bit bus mode, only BE[0]* is used. BWE[3:0]* indicate a valid data position on the data bus DATA[31:0] during write bus operation. In 16-bit bus mode, only BWE[1:0]* are used. In 8-bit bus mode, only BWE[0]* is used. The following shows the correspondence between BE[3:0]*/BWE[3:0]* and the data bus signals. BE[3]*/BWE[3]*: DATA[31:24] BE[2]*/BWE[2]*: DATA[23:16] BE[1]*/BWE[1]*: DATA[15:8] BE[0]*/BWE[0]*: DATA[7:0] The boot configuration signal on the DATA[5] pin and the EBCCRn.BC bit of the external bus controller determine whether the signals are used as BE[3:0]* or BWE[3:0]*.	All High
ACK*/ READY	Input/output PU	Data Acknowledge/Ready Flow control signal (refer to Section “7.3.6 Access Modes”).	High

3.1.4 DMA Interface Signals

Table 3.1.4 DMA Interface Signals

Signal Name	Type	Description	Initial State
DMAREQ[3:0]	Input PU	DMA Request DMA transfer request signals from an external I/O device. The DMAREQ[3:1] signal shares the pin with the other function signal (refer to Section “3.3 Pin multiplex”).	Input
DMAACK[3:0]	Output	DMA Acknowledge DMA transfer acknowledge signals to an external I/O device. The DMAACK[3:1] signal shares the other function signal (refer to Section “3.3 Pin multiplex”).	All High
DMADONE*	Input/output PU	DMA Done DMADONE* is either used as an output signal that reports the termination of DMA transfer or as an input signal that causes DMA transfer to terminate.	Input

3.1.5 PCI Interface Signals

Table 3.1.5 PCI Interface Signals (1/2)

Signal Name	Type	Description	Initial State
PCICLK[5:0]	Output	PCI Clock PCI bus clock signals. When these clock signals are not used, the pins can be set to H using the PCICLK Enable field of the pin configuration register (PCFG.PCICLKEN[5:0]).	All High
PCICLKIN	Input	PCI Feedback Clock PCI feedback clock input.	Input
PCIAD[31:0]	Input/output	PCI Address and Data Multiplexed address and data bus.	Input
C_BE[3:0]	Input/output	Command and Byte Enable Command and byte enable signals.	Input
PAR	Input/output	Parity Even parity signal for PCIAD[31:0] and C_BE[3:0]*.	Input
FRAME*	Input/output	Cycle Frame Indicates that bus operation is in progress.	Input
IRDY*	Input/output	Initiator Ready Indicates that the initiator is ready to complete data transfer.	Input
TRDY*	Input/output	Target Ready Indicates that the target is ready to complete data transfer.	Input
STOP*	Input/output	Stop The target sends this signal to the initiator to request termination of data transfer.	Input
LOCK*	Input	Lock Indicates that the PCI bus master is locking (exclusively accessing) a specified memory target on the PCI bus.	Input
ID_SEL	Input	Initialization Device Select Chip select signal used for configuration access. This pin is not used in host mode. When the PCI Controller is configured in host mode, this pin must be pulled down.	Input
DEVSEL*	Input/output	Device Select The target asserts this signal in response to access from the initiator.	Input

Table 3.1.5 PCI Interface Signals (2/2)

Signal Name	Type	Description	Initial State
REQ[3:2]*	Input	Request Signals used by the master to request bus mastership. The boot configuration signal on the DATA[2] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, REQ[3:2]* are PCI bus request input signals. In external arbiter mode, REQ[3:2]* are not used. Because the pins are still placed in the input state, they must be pulled up externally.	Input
REQ[1]* /INTOUT	Input/output/ OD	Request Signal used by the master to request bus mastership. The boot configuration signal on the DATA[2] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, this signal is a PCI bus request input signal. In external arbiter mode, this signal is an external interrupt output signal (INTOUT). Refer to Section “15.3.7 Interrupt Requests”.	Selected by DATA[2] H: Input L: Hi-Z
REQ[0]*	Input/output	Request Signal used by the master to request bus mastership. The boot configuration signal on the DATA[2] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, this signal is a PCI bus request input signal. In external arbiter mode, this signal is a PCI bus request output signal.	Selected by DATA[2] H: Input L: High
GNT[3:0]*	Input/output	Grant Indicates that bus mastership has been granted to the PCI bus master. The boot configuration signal on the DATA[2] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, all of GNT[3:0]* are PCI bus grant output signals. In external arbiter mode, GNT[0]* is a PCI bus grant input signal. Because GNT[3:1]* also become input signals, they must be pulled up externally.	Selected by DATA[2] H: All High L: Input
PERR*	Input/output	Data Parity Error Indicates a data parity error in a bus cycle other than special cycles.	Input
SERR*	Input/OD	System Error Indicates an address parity error, a data parity error in a special cycle, or a fatal error. In host mode, SERR* is an input signal. In satellite mode, SERR* is an open-drain output signal. The mode is determined by the boot configuration signal on the ADDR[19] pin.	Input
M66EN	Input/output	PCI Bus 66 MHz Clock Enable 1: Enable 66 MHz operating mode. 0: Disable 66 MHz operating mode. This pin is configured as input in satellite mode and as output in host mode. The mode is selected through the logic level of the ADDR[19] pin at boot time. This pin must be pulled down when the PCI Controller is configured in satellite mode and when the 66-MHz operating mode is disabled.	Selected by ADDR[19] H: Low L: Input
PME*	Input/OD	Power Management Event PME* indicates the power management mode. In host mode, PME* is an input signal. In satellite mode, PME* is an open-drain output signal. The mode is determined by the boot configuration signal on the ADDR[19] pin.	Selected by ADDR[19] H: Input L: Hi-Z
EEPROM_DI	Input PU	EEPROM Data In Data input from serial EEPROM for initially setting the PCI configuration.	Input
EEPROM_DO	Output	EEPROM Data Out Data output to serial EEPROM for initially setting the PCI configuration.	Low
EEPROM_CS	Output	EEPROM Chip Select Chip select for serial EEPROM for initially setting the PCI configuration.	Low
EEPROM_SK	Output	EEPROM Serial Clock Clock for serial EEPROM for initially setting the PCI configuration.	Low

Note: The PCI bus specification specifies that the following pins require pullups: FRAME*, IRDY*, TRDY*, STOP*, LOCK*, DEVSEL*, PERR*, SERR* and PME*. If these pins are unused, pullups must be provided externally to the TX4938.

3.1.6 Serial I/O Interface Signals

Table 3.1.6 Serial I/O Interface Signals

Signal Name	Type	Description	Initial State
CTS [1:0]*	Input PU	SIO Clear to Send CTS* signals. CTS[1]* share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Input
RTS [1:0]*	Output	SIO Request to Send RTS* signals. RTS[1]* share pins with other function signals (refer to Section “3.3 Pin multiplex”).	All Low
RXD[1:0]	Input PU	SIO Receive Data Serial data input signals. RXD[1] share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Input
TXD[1:0]	3-state Output	SIO Transmit Data Serial data output signals. TXD[1] share pins with other function signals (refer to Section “3.3 Pin multiplex”).	All High
SCLK	Input PU	External Serial Clock SIO clock input signal. SIO0 and SIO1 share this signal.	Input

3.1.7 Timer Interface Signals

Table 3.1.7 Timer Interface Signals

Signal Name	Type	Description	Initial State
TIMER[1:0]	Output	Timer Output Timer output signals.	All High
TCLK	Input PU	External Timer Clock Timer input clock signal. TMR0, TMR1, and TMR2 share this signal.	Input
WDRST*	OD output	Watchdog Reset Watchdog reset output signal.	Hi-Z

3.1.8 Parallel I/O Interface Signals

Table 3.1.8 Parallel I/O Interface Signals

Signal Name	Type	Description	Initial State
PIO[15:8]	Input/output PU	PIO Ports[15:8] Parallel I/O signals. PIO[15:8] share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Input
PIO[7:0]	Input/output PU	PIO Ports[7:0] Parallel I/O signals. PIO[7:0] share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Input

3.1.9 AC-link Interface Signals

Table 3.1.9 AC-link Interface Signals

Signal Name	Type	Description	Initial State
ACRESET*	Output	AC '97 Master H/W Reset ACRESET* share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Low
SYNC	Output	48 kHz Fixed Rate Sample Sync SYNC share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Low
SDOUT	Output PU	Serial, Time Division Multiplexed, AC '97 Output Stream SDOUT shares share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Low
SDIN[1]	Input	Serial, Time Division Multiplexed, AC ‘97 Input Stream SDIN[1] share pins with other function signals (refer to Section “3.3 Pin multiplex”). When this pin is used as SDIN[1], pull down by the resister on the board. (Regarding the value of register, please ask the Engineering Department in Toshiba).	Input
SDIN[0]	Input PU	Serial, Time Division Multiplexed, AC '97 Input Stream SDIN[0] share pins with other function signals (refer to Section “3.3 Pin multiplex”). When this pin is used as SDIN[0], pull down by the resister on the board. (Regarding the value of register, please ask the Engineering Department in Toshiba).	Input
BITCLK	Input PU	12.288 MHz Serial Data Clock BITCLK share pins with other function signals (refer to Section “3.3 Pin multiplex”). When this pin is used as BITCLK, pull down by the resister on the board. (Regarding the value of register, please ask the Engineering Department in Toshiba).	Input

3.1.10 Interrupt Signals

Table 3.1.10 Interrupt Signals

Signal Name	Type	Description	Initial State
NMI*	Input PU	Non-Maskable Interrupt Non-maskable interrupt signal.	Input
INT[5:0]	Input PU	External Interrupt Requests External interrupt request signals. INT[4:3] share pins with other function signals (refer to Section “3.3 Pin multiplex”).	Input

3.1.11 SPI Interface Signals

Table 3.1.11 SPI Interface Signals

Signal Name	Type	Description	Initial State
SPICLK	Input PU	SPI Clock This pin is used for a data clock to or from an SPI slave device. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
SPIOUT	Output	SPI Data Output This signal contains data to be shifted to an SPI slave device. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	High
SPIIN	Input PU	SPI Data Inputxternal Interrupt Requests This signal contains data to be shifted from an SPI slave device. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input

3.1.12 ISA/ATA Interface Signals

Table 3.1.12 SPI Interface Signals

Signal Name	Type	Description	Initial State
IOR*	Output PU	IO read This is IO read signal. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
IOW*	Output	IO write This is IO write signal. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	High
CS[1:0]*	Output PU	ATA chip select This is ATA chip select signal. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input

3.1.13 ETHER Interface Signals (Channel 0)

Table 3.1.13 ETHER Interface Signals (Channel 0)

Signal Name	Type	Description	Initial State
E0COL	Input	Collision detection signal This is collision detection signal. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E0TXCLK	Input PU	Transmit clock This is IO write signal. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E0TXD[3:0]	Output PU	Transmit data This is the transmitting data signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Low
E0TXEN	Output PU	Transmit Enable This is the transmitting enable signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Low
E0TXER	Output PU	Transmit Error indicator This is the transmitting error signal. These signals are common with the other funxtions (refer to Section “3.3 Pin multiplex”).	Low
E0CRS	Input PU	Carrier sense signal This is the carrier sense signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E0RXCLK	Input PU	Receive clock This is the receive clock signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E0RXD[3:0]	Input PU	Receive data This is the receive data signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E0RXDV	Input PU	Receive data valid signal This is the receive data valid signal. These signals are common with the other functions(refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E0RXER	Input PU	Receive Error indicator input This is the receive error indicator signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E0MDC	Output	Management data clock output This is the management data clock signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Low
E0MDIO	Input/output	Management data bi-directional signal This is the management data bi-directional signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input

3.1.14 ETHER Interface Signals (Channel 1)

Table 3.1.14 ETHER Interface Signals (Channel 1)

Signal Name	Type	Description	Initial State
E1COL	Input	Collision detection signal This is collision detection signal. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E1TXCLK	Input	Transmit clock This is IO write signal. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E1TXD[3:0]	Output	Transmit data This is the transmitting data signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Low
E1TXEN	Output	Transmit Enable This is the transmitting enable signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Low
E1TXER	Output	Transmit Error indicator This is the transmitting error signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”).	Low
E1CRS	Input	Carrier sense signal This is the carrier sense signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E1RXCLK	Input	Receive clock This is the receive clock signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E1RXD[3:0]	Input	Receive data This is the receive data signal. These signals are common with the other functions(refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E1RXDV	Input PU	Receive data valid signal This is the receive data valid signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E1RXER	Input	Receive Error indicator input This is the receive error indicator signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
E1MDC	Output	Management data clock output This is the management data clock signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Low
E1MDIO	Input/output PU	Management data bi-directional signal This is the management data bi-directional signal. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input

3.1.15 NAND Flash Memory Interface Signals

Table 3.1.15 NAND Flash Memory Interface Signals

Signal Name	Type	Description	Initial State
ND_ALE	Output PU	NAND Flash Address Latch Enable ALE signal for NAND flash memory. These signals are common with the other functions (refer to Section “3.3 Pin Multiplex”). Use the configuration setting during boot-up.	Input
ND_CLE	Output	NAND Flash Command Latch Enable CLE signal for NAND flash memory. These signals are common with the other functions (refer to Section “3.3 Pin Multiplex”). Use the configuration setting during boot-up.	High
ND_CE*	Output	NAND Flash Chip Enable CE signal for NAND flash memory. Refer to Section “3.3 Pin multiplex”	High
ND_RE*	Output PU	NAND Flash Read Enable RE signal for NAND flash memory. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input
ND_WE*	Output	NADND Flash Write Enable WE signal for NAND flash memory. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	High
ND_RB*	Input PU	NAND Flash Ready/Busy Ready/Busy signal for NAND flash memory. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Input

3.1.16 Extended EJTAG Interface Signals

Table 3.1.16 Extended EJTAG Interface Signals (1/2)

Signal Name	Type	Description	Initial State
TCK	Input PU	JTAG Test Clock Input Clock input signal for JTAG. TCK is used to execute JTAG instructions and input/output data.	Input
TDI/DINT*	Input PU	JTAG Test Data Input/Debug Interrupt When PC trace mode is not selected, this signal is a JTAG data input signal. It is used to input serial data to JTAG data/instruction registers. When PC trace mode is selected, this signal is an interrupt input signal used to cancel PC trace mode for the debug unit.	Input
TDO/TPC[0]	Output	JTAG Test Data Output/PC Trace Output When PC trace mode is not selected, this signal is a JTAG data output signal. Data is output by means of serial scan. When PC trace mode is selected, this signal outputs the value of the non-continuous program counter in sync with the debug clock (DCLK).	High
TPC[3:1]	Output	PC Trace Output TPC[3:1] output the value of the noncontiguous program counter in sync with DCLK. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	All High
TMS	Input PU	JTAG Test Mode Select Input TMS mainly controls state transition in the TAP controller state machine.	Input
TRST*	Input	Test Reset Input Asynchronous reset input for the TAP controller and debug support unit (DSU). TRST* pin must be pulled down (ex. 10 kΩ). When this signal is deasserted, G-Bus timeout detection is disabled (refer to Section “5.1.1 Detecting G-Bus Timeout”).	Input

Table 3.1.16 Extended EJTAG Interface Signals (2/2)

Signal Name	Type	Description	Initial State
DCLK	Output	Debug Clock Clock output signal for the real-time debugging system. When PC trace mode is selected, the TPC[3:1] and PCST signals are output synchronously. This clock is the TX49/H3 core operating clock (CPUCLK) divided by 3. This signal is common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	Low
PCST[8:0]	Output	PC Trace Status Information Outputs PC trace status and other information. These signals are common with the other functions (refer to Section “3.3 Pin multiplex”). Use the configuration setting during boot-up.	All Low

3.1.17 Clock Signals

Table 3.1.17 Clock Signals

Signal Name	Type	Description	Initial State
MASTERCLK	Input	Master Clock Input pin for the TX4938 operating clock. A crystal resonator cannot be connected to this pin because the pin does not contain an oscillator.	Input
HALTDOZE	Output	Halt/Doze State Output This signal is asserted (High output) when the TX4938 enters Halt or Doze mode.	Low
BYPASSPLL*	Input	Bypass PLL This pin must be fixed to High.	Input
CGRESET*	Input	CG Reset CGRESET* initializes the CG.	Input

3.1.18 Initialization Signal

Table 3.1.18 Initialization Signal

Signal Name	Type	Description	Initial State
RESET*	Input	Reset Reset signal.	Input

3.1.19 Test Signals

Table 3.1.19 Test Signals

Signal Name	Type	Description	Initial State
TEST[4:0]*	Input PU	Test Mode Setting Test pins. These pins must be left open or fixed to High. TEST[1]* may be used when debugging the system. Toshiba recommends that your board design enable the pin to be driven low after the TX4938 is mounted on the PC board. Contact Toshiba technical staff for more information on the TEST[1]* functions.	Input

3.1.20 Power Supply Pins

Table 3.1.20 Power Supply Pins

Signal Name	Type	Description	Initial State
PLL1VDD_A, PLL2VDD_A	—	PLL Power Pins PLL analog power supply pins. PLL1VDD_A = 1.5 V. PLL2VDD_A = 1.5 V.	—
PLL1VSS_A, PLL2VSS_A	—	PLL Ground Pins PLL analog ground pins. PLL1VSS_A = 0 V. PLL2VSS_A = 0 V.	—
VccInt	—	Internal Power Pins Digital power supply pins for internal logic. VccInt = 1.5 V.	—
VccIO	—	I/O Power Pins Digital power supply pins for input/output pins. VccIO = 3.3 V.	—
Vss	—	Ground Pins Digital ground pins. Vss = 0 V.	—

3.2 Boot Configuration

The ADDR[19:0] and DATA[15:0] signals can also function as configuration signals for initially setting various functions upon booting the system. The states of the configuration signals immediately after the RESET* or CGRESET* signal is deasserted are read as initial values for the TX4938 internal registers. A High signal level sets a value of 1 and a Low signal level sets a value of 0.

All configuration signals are provided with internal pull-up resistors. To drive a signal Low, pull down the corresponding pin on the board using an approx. 4.7 kΩ resistor. Driving a signal High does not require a pull-down resistor. Any signals defined as Reserved should not be pulled down.

Table 3.2.1 lists the functions that can be set using configuration signals. Table 3.2.2 and Table 3.2.3 describe each configuration signal.

Table 3.2.1 Functions that Can be Set Using Configuration Signals

Peripheral Function	Functions that Can be Set	Configuration Signal
PCI controller	PCI controller operating mode (satellite or host)	ADDR[19]
	Division ratio of PCICLK[5:0] to CPUCLK	ADDR[11:10]
	PCI bus arbiter selection (internal or external)	DATA[2]
External bus controller	Division ratio of SYSCLK to GBUSCLK	ADDR[14:13]
	Boot device selection Division ratio of the external bus controller clock upon booting	ADDR[8:6]
	BE[3:0]*/BWE[3:0]* function selection upon booting	DATA[5]
	Handling of the ACK signal upon booting (internal or external)	DATA[4]
	Data bus width for the boot device	DATA[1:0]
Ethernet controller	Operation frequency for System interface signal (Internal PCI bus)	ADDR[16]
Clock	Division ratio of CPUCLK to MASTERCLK	ADDR[3:0]
Others	Shared pin function setting	ADDR[18], ADDR[9] DATA[6], DATA[3]
	Endian setting	ADDR[12]
	Board information setting	DATA[15:8]
	Controlling built-in timer interrupts of the TX49/H3 core	DATA[7]

Table 3.2.2 Boot Configuration Specified with the ADDR[19:0] Signals (1/2)

Signal	Description	Corresponding Register Bit	Configuration Determined at
ADDR[19]	PCI Controller Mode Select Specifies the operating mode of the TX4938 PCI controller. L = Satellite H = Host	CCFG. PCIMODE	RESET* deassert edge
ADDR[18]	Select Shared I/O Pins Specifies the function of the PIO[15:8]/CB[7:0] shared pins. L = PIO[15:8] H = CB[7:0]	PCFG. SEL1	RESET* deassert edge
ADDR[17]	Reserved Used for testing. This signal will not be set to 0 upon booting.	—	—
ADDR[16]	Select internal PCI1 clock Frequency (PCIDMD) Specifies the operating mode of the TX4938 internal PCI controller for Ethernet Controller. L = PCI1CLK = GBUSCLK frequency/2 H = PCI1CLK = GBUSCLK frequency/4	CCFG. PCI1DMD	CGRESET* deassert edge
ADDR[15]	Reserved Used for testing. This signal will not be set to 0 upon booting.	—	—
ADDR[14:13]	Select SYSCLK Frequency Specifies the division ratio of the SYSCLK frequency to the G-Bus clock (GBUSCLK) frequency. LL = 4 (SYSCLK frequency = GBUSCLK frequency/4) LH = 3 (SYSCLK frequency = GBUSCLK frequency/3) HL = 2 (SYSCLK frequency = GBUSCLK frequency/2) HH = 1 (SYSCLK frequency = GBUSCLK frequency)	CCFG. SYSSP	CGRESET* deassert edge
ADDR[12]	TX4938 Endian Mode Specifies the TX4938 endian mode. L = Little endian H = Big endian	CCFG. ENDIAN	RESET* deassert edge
ADDR[11:10]	Select PCI Clock Frequency Specifies the division ratio of the PCI bus clock (PCICLK[5:0]) to the TX49/H3 core clock (CPUCLK). Initial value of CCFG[12] is 0. LL = 8 (PCICLK frequency = CPUCLK frequency/8) LH = 9 (PCICLK frequency = CPUCLK frequency/9) HL = 10 (PCICLK frequency = CPUCLK frequency/10) HH = 11 (PCICLK frequency = CPUCLK frequency/11)	CCFG. PCIDIVMODE	CGRESET* deassert edge
ADDR[9]	PIO[4:2]/ACLC/DMAREQ[2]/DMAACK[2] Select Specifies whether PIO[4:2]/DMAREQ[2]/DMAACK[2] signals are used as PIO or AC-link interface signals. L = PIO H = AC-link interface	PCFG.SEL2	RESET* deassert edge
ADDR[8:6]	Select Boot Memory and Device Clock Frequency Specifies the clock division ratio for external bus controller channel 0 upon booting (refer to Section “7.3.8 Clock Options”). HHH = Device connected to channel 0 of the external bus controller (Clock division rate = 1/1) HHL = Device connected to channel 0 of the external bus controller (Clock division rate = 1/2) HLH = Device connected to channel 0 of the external bus controller (Clock division rate = 1/3) HLL = Device connected to channel 0 of the external bus controller (Clock division rate = 1/4) LHH =PCI boot LHL = NAND Flash memory boot (NAND IPL) LLH and LLL = Reserved	ADDR[8]: EBCCR0.ME ADDR[7:6] EBCCR0.SP	RESET* deassert edge

Table 3.2.2 Boot Configuration Specified with the ADDR[19:0] Signals (2/2)

Signal	Description	Corresponding Register Bit	Configuration Determined at
ADDR[5]	Select SDRAM device Select initial setting derivability of SDRAM interface signals L = 8mA H = 16mA ADDR[19:0], CKE, RAS*, CAS*, WE*, SDCS[3:0], SDCLK[3:0], SDCLKIN	PCFG[53:40]	RESET* deassert edge
ADDR[4]	Initial Setting of Boot Select initial setting derivability of SDRAM interface signals L = 8mA H = 16mA DATA[63:0], CB[7:0], DQM[7:0]	PCFG[56:54]	RESET* deassert edge
ADDR[3:0]	CPUCLK Clock Speed Setting Specifies the value by which MASTERCLK input signal is multiplied to produce the TX49/H3 core clock (CPUCLK). The values of ADDR[1:0] are also reflected in the EC field of the TX49/H3 core Config register. ADDR[3:0] : DIVMODE[3:0] HHHH:0100 CPUCLK frequency = 2 x MASTERCLK frequency HHHL:1111 CPUCLK frequency = 2.5 x MASTERCLK frequency HHLH:0101 CPUCLK frequency = 3 x MASTERCLK frequency HHLL:0110 CPUCLK frequency = 4 x MASTERCLK frequency LHHH:1101 CPUCLK frequency = 4.5 x MASTERCLK frequency LHHL: - reserved LHLH: - reserved LHLL: - reserved HLHH:0000 CPUCLK frequency = 8 x MASTERCLK frequency HLHL:1011 CPUCLK frequency = 10 x MASTERCLK frequency HLLH:0001 CPUCLK frequency = 12 x MASTERCLK frequency HLLL:0010 CPUCLK frequency = 16 x MASTERCLK frequency LLHH:1001 CPUCLK frequency = 18 x MASTERCLK frequency LLHL: - reserved LLLH: - reserved LLLL: - reserved	CCFG.DIVMODE	CGRESET* deassert edge

Table 3.2.3 Boot Configuration Specified with the DATA[22:0] Signals

Signal	Description	Corresponding Register Bit	Configuration Determined at
DATA[22:16]	Reserved	—	—
DATA[15:8]	Boot Configuration Reads the board information and accordingly sets the boot configuration field (BCFG) of the chip configuration register (CCFG).	CCFG.BCFG	RESET* deassert edge
DATA[7]	TX49/H3 Internal Timer Interrupt Disable Specifies whether timer interrupts within the TX49/H2 core are enabled. H = Enable timer interrupts within the TX49/H3 core. L = Disable timer interrupts within the TX49/H3 core.	CCFG.TINTDIS	RESET* deassert edge
DATA[6]	L = ETHER1 usable H = ETHER1 unusable	PCFG.ETH1_SEL	RESET* deassert edge
DATA[5]	Specifies the function of the BE[3:0]*/BWE[3:0]* pins upon booting. L = BE[3:0]* (Byte Enable) H = BWE[3:0]* (Byte Write Enable)	EBCCR0.BC	RESET* deassert edge
DATA[4]	Boot ACK* Input Specifies the access mode for external bus controller channel 0. L = External ACK mode H = Normal mode	EBCCR0.WT[0]	RESET* deassert edge
DATA[3]	L = ETHER0 usable H = ETHER0 unusable	PCFG.ETH0_SEL	RESET* deassert edge
DATA[2]	PCI Arbiter Select Selects a PCI bus arbiter. L = External PCI bus arbiter. H = Built-in PCI bus arbiter.	CCFG.PCIARB	RESET* deassert edge
DATA[1:0]	Boot ROM Bus Width Specifies the data bus width when booting from a memory device connected to the external bus controller. LL = Reserved LH = 32 bits HL = 16 bits HH = 8 bits	EBCCR0.BSZ	RESET* deassert edge

3.3 Pin multiplex

The TX4938 has some multiplexed pins. Each pin is used for different functions depending on the settings of the PCFG[61:58] control register and ADDR[18]/[9], DATA[6]/[3] boot configuration signal. Table 3.3.1 shows how to set the function for each pin.

Table 3.3.1 Pin multiplex (1/2)

	Signal, PCFG	Function on pin-multiplex								
		ECC	PIO[15:8]	ACLC	ETHER0	ETHER1	ATA/ISA	ISA	SPI	NDFMC
Setting of Boot config. signal and PCFG (Note1) (Note2)	ADDR[18]	H	L	*	H	*	*	*	*	*
	ADDR[9]	*	*	H	L	*	*	*	*	*
	DATA[6]	*	*	*	*	L	*	*	*	*
	DATA[3]	H	H	H	L	*	*	*	*	*
	PCFG[61]	*	*	*	*	*	1	0	*	0
	PCFG[60]	*	*	*	*	*	*	1	*	0
	PCFG[59]	*	*	*	*	*	*	*	1	*
	PCFG[58]	*	*	*	*	*	0	0	*	1
Function of Each signal (Note3) (Note4)	CE[7]					E1COL				
	CE[6]					E1CRS				
	CE[5]					E1TXER				
	CE[4]									ND_CE*
	CE[3]									ND_CLE
	CB[7]	CB[7]	PIO[15]		E0TXD[3]					
	CB[6]	CB[6]	PIO[14]		E0TXD[2]					
	CB[5]	CB[5]	PIO[13]		E0RXD[3]					
	CB[4]	CB[4]	PIO[12]		E0RXD[2]					
	CB[3]	CB[3]	PIO[11]		E0TXD[1]					
	CB[2]	CB[2]	PIO[10]		E0TXD[0]					
	CB[1]	CB[1]	PIO[9]		E0RXD[1]					
	CB[0]	CB[0]	PIO[8]		E0RXD[0]					
	DMAREQ[3]						IOR*	IOR*		ND_RE*
	DMAACK[3]						IOW*	IOW*		ND_WE*
	DMAREQ[2]			ACRESET*	E0TXCLK					
	DMAACK[2]			SYNC	E0MDIO					
	DMAREQ[1]								SPIN	
	DMAACK[1]								SPIOUT	
	DMAREQ[0]									
	DMAACK[0]									
	PIO[7]									
	PIO[6]						CS[1]*			ND_RB*
	PIO[5]						CS[0]*			ND_ALE
	PIO[4]			SDOUT						
	PIO[3]			SDIN[0]	E0TXEN					
	PIO[2]			BITCLK	E0CRS					

Table 3.3.1 Pin multiplex (2/2)

	Signal, PCFG	Function on pin-multiplex								
		ECC	PIO[15:8]	ACLC	ETHER0	ETHER1	ATA	ISA	SPI	NDFMC
Function of Each signal (Note3) (Note4)	PIO[1]					E1MDIO				
	PIO[0]								SPICLK	
	INT[5]									
	INT[4]				E0RXER					
	INT[3]					E1RXDV				
	INT[2]									
	INT[1]									
	INT[0]									
	CTS[1]*				E0RXCLK					
	RTS[1]*				E0MDC					
	RXD[1]				E0RXDV					
	TXD[1]				E0TXER					
	CTS[0]*									
	RTS[0]*									
	RXD[0]									
	TXD[0]									
	SDIN[1]			SDIN[1]	E0COL					
	DCLK					E1TXD[3]				
	PCST[8]					E1TXD[2]				
	PCST[7]					E1TXD[1]				
	PCST[6]					E1TXD[0]				
	PCST[5]					E1RXD[3]				
	PCST[4]					E1RXD[2]				
	PCST[3]					E1RXD[1]				
	PCST[2]					E1RXD[0]				
	PCST[1]					E1RXCLK				
	PCST[0]					E1TXCLK				
	TPC[3]					E1RXER				
	TPC[2]					E1MDC				
	TPC[1]					E1TXEN				

Note 1: * shows that there is no relationship to the configuration of the corresponding function.

Note 2: One of ECC, PIO[15:8] and ETHERC0 can be selected (can't use simultaneously).
One of ACLC and ETHERC0 can be selected (can't use simultaneously).
One of ATA and NDFMC can be selected (can't use simultaneously).
One of ISA and NDFMC can be selected (can't use simultaneously).
(Operation is not guaranteed when these functions that cannot be used simultaneously are set up)

Note 3: It shows that these terminals are multiplexed under each function.

Note 4: Blank shows that the terminal isn't multiplexed with the function.
For example, when only two pieces, ATA and SPI are selected, functions other than DMAREQ[3], DMAREQ[1], DMAACK[3], DMAACK[1], PIO[0] can be used.

4. Address Mapping

This chapter explains the physical address map of TX4938.

Please refer to "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture" about the details of mapping to a physical address from the virtual address of TX49/H3 core.

4.1 TX4938 Physical Address Map

TX4938 supports up to 64G (2^{36}) bytes of physical address.

Following resources are to be allocated in the physical address of the TX4938.

- TX4938 Internal registers (refer to “4.2 Register Map”)
- SDRAM (refer to “9.3.2 Address Mapping”)
- External Devices such as ROM, I/O Devices (refer to “7.3.3 Address Mapping”)
- PCI Bus (refer to “10.3.4 Initiator Access”)

Each resource is to be allocated in any physical addresses by the register setup. Refer to the explanation of each controller for the details of the mapping.

At initialization, only the internal registers and the memory space which stores the TX49/H3 core reset vectors are allocated shown as Figure 4.1.1. Usually ROM connected to the external bus controller channel 0 is used for the memory device that stores the reset vectors. TX4938 also supports using the memories on PCI bus as the memory device stores the reset vectors. Refer to “10.3.13 PCI Boot” for detail about this.

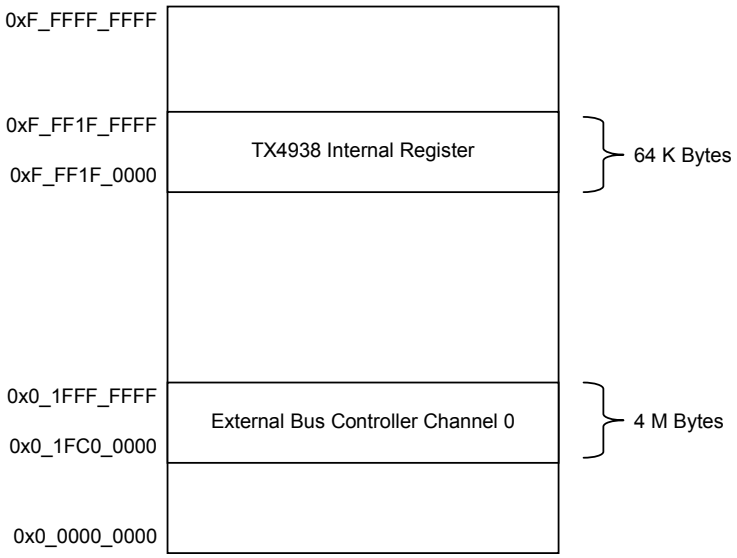


Figure 4.1.1 Physical Address Map at Initializing System

It is possible to access a resource of TX4938 as a PCI target device through PCI bus. About how to allocate resources of TX4938 to the PCI bus address space, refer to “10.3.5 Target Access”.

4.2 Register Map

4.2.1 Addressing

TX4938 internal registers are to be accessed through 64 K bytes address space that is based on physical address 0xF_FF1F_0000 or pointed address by RAMP register (refer to 5.2.7). Figure 4.2.1 shows how to generate internal register address. Physical address 1 and physical address 2 shown Figure 4.2.1 access the same register.

In TX49/H3 Core, the physical address form 0xF_FF00_0000 to 0xF_FF3F_FFFF are uncached mapped to the virtual address form 0xFF00_0000 to 0xFF3F_FFFF (32 bit mode) /form 0xFFFF_FFFF_FF00_0000 to 0xFFFF_FFFF_FF3F_FFFF (64 bit mode).

This space includes the region form 0xF_FF1F_0000 allocated TX4938 internal registers at initialization.

Figure 4.2.1 Generating Physical Address for a Internal Register

4.2.2 Ways to Access to Internal Registers

3 ways to access to the internal registers of TX4938 are supported. First is 32-bit register access. Second is 64-bit register access. Last is PCI configuration register access in PCI satellite mode.

32-bit register supports 32-bit size access only. Another size access without 32-bit size is undefined.

64-bit register supports both 64-bit size access and two times 32-bit size access. In each Endian mode, 32-bit size access is performed shown as Table 4.2.1.

When the build-in PCI controller works in the satellite mode (refer to "10.3.1 Terminology Explanation"), PCI configuration registers are to be accessed through PCI bus in configuration cycles. It is possible to access to the arbitrary size of PCI configuration register as always Little Endian space regardless the system setup.

Table 4.2.1 32-bit Size Access to 64-bit Register

Address	Big Endian (Bit which are accessed)	Little Endian (Bit which are accessed)
0x*_****_***0	[63.....32] [31.....0] ########	[63.....32] [31.....0] ########
0x*_****_***8		
0x*_****_***4	[63.....32] [31.....0] ########	[63.....32] [31.....0] ########
0x*_****_***C		

(######## means 32 bits data (upper 32 bits or lower 32 bits) which are accessed.)

4.2.3 Register Map

Please refer to “10.5 PCI Configuration Space Register” about PCI configuration register.

Table 4.2.2 Register Map

Offset Address	Peripheral Controller	Detail
0x0000 to 0x4FFF	Reserved	—
0x5000 to 0x5FFF	NDFMC	Refer to “20.4”
0x6000 to 0x67FF	SRAMC	Refer to “21.4”
0x6800 to 0x6FFF	Reserved	—
0x7000 to 0x7FFF	PCIC1	Refer to “16” (For ETHERC0 and ETHERC1)
0x8000 to 0x8FFF	SDRAMC	Refer to “9.4”
0x9000 to 0x9FFF	EBUSC	Refer to “7.4”
0xA000 to 0xAFFF	ECC	Refer to “9.4”
0xB000 to 0xB7FF	DMAC0	Refer to “8.4”
0xB800 to 0xBFFF	DMAC1	Refer to “8.4”
0xD000 to 0xDFFF	PCIC	Refer to “10.4”
0xE000 to 0xEFFF	CONFIG	Refer to “5.2”
0xF000 to 0xF0FF	TMR0	Refer to “12.4”
0xF100 to 0xF1FF	TMR1	Refer to “12.4”
0xF200 to 0xF2FF	TMR2	Refer to “12.4”
0xF300 to 0xF3FF	SIO0	Refer to “11.4”
0xF400 to 0xF4FF	SIO1	Refer to “11.4”
0xF500 to 0xF50F	PIO	Refer to “13.4”
0xF510 to 0xF6FF	IRC	Refer to “15.4”
0xF700 to 0xF7FF	ACLC	Refer to “14.4”
0xF800 to 0xF8FF	SPIC	Refer to “17.4”
0xF900 to 0xFFFF	Reserved	

Table 4.2.3 Internal Registers (1/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
NAND Flash Memory Controller (NDFMC)			
0x5000	64	NDFDTR	NAND Flash Memory Data Transfer Register
0x5008	64	NDFMCR	NAND Flash Memory Mode Control Register
0x5010	64	NDFSR	NAND Flash Memory Status Register
0x5018	64	NDFISR	NAND Flash Memory Interrupt Status Register
0x5020	64	NDFIMR	NAND Flash Memory Interrupt Mask Register
0x5028	64	NDFSPR	NAND Flash Memory Strobe Pulse Width Register
0x5030	64	NDFRSTR	NAND Flash Memory Reset Register
SRAM Controller (SRAMC)			
0x6000	64	SRAMCR	Internal SRAM Control Register

Table 4.2.3 Internal Registers (2/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
PCI Controller for ETHERC (PCIC1)			
0x7000	32	PCIID	ID Register (Device ID, Vender ID)
0x7004	32	PCISTATUS	PCI Status Command Register (Status, Command)
0x7008	32	PCICCREV	Class Code Revision ID Register (Class Code, Revision ID)
0x700C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
0x7010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
0x7014	32	P2GM0PUBASE	P2G Memory Space 0 PCI Upper Base Address Register (Base Address 0 Upper)
0x7018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
0x701C	32	P2GM1PUBASE	P2G Memory Space 1 PCI Upper Base Address Register (Base Address 1 Upper)
0x7020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
0x7024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
0x702C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vender ID)
0x7034	32	PCICAPPTR	Capability Pointer Register (Capability Register)
0x703C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
0x7040	32	G2PTOCNT	G2P Timeout Count Register (Retry Timeout Value, TRDY Timeout Value)
0x7080	32	G2PSTATUS	G2P Status Register
0x7084	32	G2PMASK	G2P Interrupt Mask Register
0x7088	32	PCISSTATUS	Satellite Mode PCI Status Register (Status, PMCSR)
0x708C	32	PCIMASK	PCI Status Interrupt Mask Register
0x7090	32	P2GCFG	P2G Configuration Register
0x7094	32	P2GSTATUS	P2G Status Register
0x7098	32	P2GMASK	P2G Interrupt Mask Register
0x709C	32	P2GCCMD	P2G Current Command Register
0x7100	32	PBAREQPORT	PCI Bus Arbiter Request port Register
0x7104	32	PBACFG	PCI Bus Arbiter Configuration Register
0x7108	32	PBASTATUS	PCI Bus Arbiter Status Register
0x710C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
0x7110	32	PBABM	PCI Bus Arbiter Broken Master Register
0x7114	32	PBACREQ	PCI Bus Arbiter Current Request Register
0x7118	32	PBACGNT	PCI Bus Arbiter Current Grant Register
0x711C	32	PBACSTATE	PCI Bus Arbiter Current State Register

Table 4.2.3 Internal Registers (3/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
PCI Controller for ETHERC (PCIC1)			
0x7120	64	G2PM0GBASE	G2P Memory Space 0 G-bus Base Address Register
0x7128	64	G2PM1GBASE	G2P Memory Space 1 G-bus Base Address Register
0x7130	64	G2PM2GBASE	G2P Memory Space 2 G-bus Base Address Register
0x7138	64	G2PIOGBASE	G2P I/O Space G-bus Base Address Register
0x7140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
0x7144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
0x7148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
0x714C	32	G2PIOMASK	G2P I/O Space Address Mask Register
0x7150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
0x7158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
0x7160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
0x7168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
0x7170	32	PCICCFG	PCI Controller Configuration Register
0x7174	32	PCICSTATUS	PCI Controller Status Register
0x7178	32	PCICMASK	PCI Controller Interrupt Mask Register
0x7180	64	P2GM0GBASE	P2G Memory Space 0 G-bus Base Address Register
0x7188	64	P2GM1GBASE	P2G Memory Space 1 G-bus Base Address Register
0x7190	64	P2GM2GBASE	P2G Memory Space 2 G-bus Base Address Register
0x7198	64	P2GIOGBASE	P2G I/O Space G-bus Base Address Register
0x71A0	32	G2PCFGADRS	G2P Configuration Address Register
0x71A4	32	G2PCFGDATA	G2P Interrupt Acknowledge Data Register
0x71C8	32	G2PINTACK	G2P Interrupt Acknowledge Data Register
0x71CC	32	G2PSPC	G2P Special Cycle Data Register

Table 4.2.3 Internal Registers (4/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
SDRAM Controller (SDRAMC)			
0x8000	64	SDCCR0	SDRAM Channel Control Register 0
0x8008	64	SDCCR1	SDRAM Channel Control Register 1
0x8010	64	SDCCR2	SDRAM Channel Control Register 2
0x8018	64	SDCCR3	SDRAM Channel Control Register 3
0x8040	64	SDCTR	SDRAM Timing Register
0x8058	64	SDCCMD	SDRAM Command Register
External Bus Controller (EBUSC)			
0x9000	64	EBCCR0	EBUS Channel Control Register 0
0x9008	64	EBCCR1	EBUS Channel Control Register 1
0x9010	64	EBCCR2	EBUS Channel Control Register 2
0x9018	64	EBCCR3	EBUS Channel Control Register 3
0x9020	64	EBCCR4	EBUS Channel Control Register 4
0x9028	64	EBCCR5	EBUS Channel Control Register 5
0x9030	64	EBCCR6	EBUS Channel Control Register 6
0x9038	64	EBCCR7	EBUS Channel Control Register 7
SDRAM Error Check Correction (ECC)			
0xA000	64	ECCCR	ECC Control Register
0xA008	64	ECCSR	ECC Status Register

Table 4.2.3 Internal Registers (5/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
DMA Controller (DMAC0)			
0xB000	64	DM0CHAR0	DMAC0 Chain Address Register 0
0xB008	64	DM0SAR0	DMAC0 Source Address Register 0
0xB010	64	DM0DAR0	DMAC0 Destination Address Register 0
0xB018	64	DM0CNTR0	DMAC0 Count Register 0
0xB020	64	DM0SAIR0	DMAC0 Source Address Increment Register 0
0xB028	64	DM0DAIR0	DMAC0 Destination Address Increment Register 0
0xB030	64	DM0CCR0	DMAC0 Channel Control Register 0
0xB038	64	DM0CSR0	DMAC0 Channel Status Register 0
0xB040	64	DM0CHAR1	DMAC0 Chain Address Register 1
0xB048	64	DM0SAR1	DMAC0 Source Address Register 1
0xB050	64	DM0DAR1	DMAC0 Destination Address Register 1
0xB058	64	DM0CNTR1	DMAC0 Count Register 1
0xB060	64	DM0SAIR1	DMAC0 Source Address Increment Register 1
0xB068	64	DM0DAIR1	DMAC0 Destination Address Increment Register 1
0xB070	64	DM0CCR1	DMAC0 Channel Control Register 1
0xB078	64	DM0CSR1	DMAC0 Channel Status Register 1
0xB080	64	DM0CHAR2	DMAC0 Chain Address Register 2
0xB088	64	DM0SAR2	DMAC0 Source Address Register 2
0xB090	64	DM0DAR2	DMAC0 Destination Address Register 2
0xB098	64	DM0CNTR2	DMAC0 Count Register 2
0xB0A0	64	DM0SAIR2	DMAC0 Source Address Increment Register 2
0xB0A8	64	DM0DAIR2	DMAC0 Destination Address Increment Register 2
0xB0B0	64	DM0CCR2	DMAC0 Channel Control Register 2
0xB0B8	64	DM0CSR2	DMAC0 Channel Status Register 2
0xB0C0	64	DM0CHAR3	DMAC0 Chain Address Register 3
0xB0C8	64	DM0SAR3	DMAC0 Source Address Register 3
0xB0D0	64	DM0DAR3	DMAC0 Destination Address Register 3
0xB0D8	64	DM0CNTR3	DMAC0 Count Register 3
0xB0E0	64	DM0SAIR3	DMAC0 Source Address Increment Register 3
0xB0E8	64	DM0DAIR3	DMAC0 Destination Address Increment Register 3
0xB0F0	64	DM0CCR3	DMAC0 Channel Control Register 3
0xB0F8	64	DM0CSR3	DMAC0 Channel Status Register 3
0xB148	64	DM0MFDR	DMAC0 Memory Fill Data Register
0xB150	64	DM0MCR	DMAC0 Master Control Register

Table 4.2.3 Internal Registers (6/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
DMA Controller (DMAC1)			
0xB800	64	DM1CHAR0	DMAC1 Chain Address Register 0
0xB808	64	DM1SAR0	DMAC1 Source Address Register 0
0xB810	64	DM1DAR0	DMAC1 Destination Address Register 0
0xB818	64	DM1CNTR0	DMAC1 Count Register 0
0xB820	64	DM1SAIR0	DMAC1 Source Address Increment Register 0
0xB828	64	DM1DAIR0	DMAC1 Destination Address Increment Register 0
0xB830	64	DM1CCR0	DMAC1 Channel Control Register 0
0xB838	64	DM1CSR0	DMAC1 Channel Status Register 0
0xB840	64	DM1CHAR1	DMAC1 Chain Address Register 1
0xB848	64	DM1SAR1	DMAC1 Source Address Register 1
0xB850	64	DM1DAR1	DMAC1 Destination Address Register 1
0xB858	64	DM1CNTR1	DMAC1 Count Register 1
0xB860	64	DM1SAIR1	DMAC1 Source Address Increment Register 1
0xB868	64	DM1DAIR1	DMAC1 Destination Address Increment Register 1
0xB870	64	DM1CCR1	DMAC1 Channel Control Register 1
0xB878	64	DM1CSR1	DMAC1 Channel Status Register 1
0xB880	64	DM1CHAR2	DMAC1 Chain Address Register 2
0xB888	64	DM1SAR2	DMAC1 Source Address Register 2
0xB890	64	DM1DAR2	DMAC1 Destination Address Register 2
0xB898	64	DM1CNTR2	DMAC1 Count Register 2
0xB8A0	64	DM1SAIR2	DMAC1 Source Address Increment Register 2
0xB8A8	64	DM1DAIR2	DMAC1 Destination Address Increment Register 2
0xB8B0	64	DM1CCR2	DMAC1 Channel Control Register 2
0xB8B8	64	DM1CSR2	DMAC1 Channel Status Register 2
0xB8C0	64	DM1CHAR3	DMAC1 Chain Address Register 3
0xB8C8	64	DM1SAR3	DMAC1 Source Address Register 3
0xB8D0	64	DM1DAR3	DMAC1 Destination Address Register 3
0xB8D8	64	DM1CNTR3	DMAC1 Count Register 3
0xB8E0	64	DM1SAIR3	DMAC1 Source Address Increment Register 3
0xB8E8	64	DM1DAIR3	DMAC1 Destination Address Increment Register 3
0xB8F0	64	DM1CCR3	DMAC1 Channel Control Register 3
0xB8F8	64	DM1CSR3	DMAC1 Channel Status Register 3
0xB948	64	DM1MFDR	DMAC1 Memory Fill Data Register
0xB950	64	DM1MCR	DMAC1 Master Control Register

Table 4.2.3 Internal Registers (7/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
PCI Controller (PCIC)			
0xD000	32	PCIID	ID Register (Device ID, Vendor ID)
0xD004	32	PCISTATUS	PCI Status, Command Register (Status, Command)
0xD008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
0xD00C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
0xD010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
0xD014	32	P2GM0PUBASE	P2G Memory Space 0 PCI Upper Base Address Register (Base Address 0 Upper)
0xD018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
0xD01C	32	P2GM1PUBASE	P2G Memory Space 1 PCI Upper Base Address Register (Base Address 1 Upper)
0xD020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
0xD024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
0xD02C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
0xD034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
0xD03C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
0xD040	32	G2PTOCNT	G2P Timeout Count register (Retry Timeout Value, TRDY Timeout Value)
0xD080	32	G2PSTATUS	G2P Status Register
0xD084	32	G2PMASK	G2P Interrupt Mask Register
0xD088	32	PCISSTATUS	Satellite Mode PCI Status Register (Status, PMCSR)
0xD08c	32	PCIMASK	PCI Status Interrupt Mask Register
0xD090	32	P2GCFG	P2G Configuration Register
0xD094	32	P2GSTATUS	P2G Status Register
0xD098	32	P2GMASK	P2G Interrupt Mask Register
0xD09C	32	P2GCCMD	P2G Current Command Register
0xD100	32	PBAREQPORT	PCI Bus Arbiter Request Port Register
0xD104	32	PBACFG	PCI Bus Arbiter Configuration Register
0xD108	32	PBASTATUS	PCI Bus Arbiter Status Register
0xD10C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
0xD110	32	PBABM	PCI Bus Arbiter Broken Master Register
0xD114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for a diagnosis)
0xD118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for a diagnosis)
0xD11C	32	PBACSTATE	PCI Bus Arbiter Current Status Register (for a diagnosis)

Table 4.2.3 Internal Registers (8/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
0xD120	64	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
0xD128	64	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
0xD130	64	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
0xD138	64	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
0xD140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
0xD144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
0xD148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
0xD14C	32	G2PIOMASK	G2P I/O Space Address Mask Register
0xD150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
0xD158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
0xD160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
0xD168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
0xD170	32	PCICCFG	PCI Controller Configuration Register
0xD174	32	PCICSTATUS	PCI Controller Status Register
0xD178	32	PCICMASK	PCI Controller Interrupt Mask register
0xD180	64	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
0xD188	64	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
0xD190	64	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
0xD198	64	P2GIOGBASE	P2G I/O Space 0 G-Bus Base Address Register
0xD1A0	32	G2PCFGADRS	G2P Configuration Address Register
0xD1A4	32	G2PCFGDATA	G2P Configuration Data Register
0xD1C8	32	G2PINTACK	G2P Interrupt Acknowledge Register
0xD1CC	32	G2PSPC	G2P Special Cycle Data Register
0xD1D0	32	PCICDATA0	PCI Configuration Data 0 Register
0xD1D4	32	PCICDATA1	PCI Configuration Data 1 Register
0xD1D8	32	PCICDATA2	PCI Configuration Data 2 Register
0xD1DC	32	PCICDATA3	PCI Configuration Data 3 Register
0xD200	64	PDMCA	PDMAC Chain Address Register
0xD208	64	PDMGA	PDMAC G-Bus Address Register
0xD210	64	PDMPA	PDMAC PCI Bus Address Register
0xD218	64	PDMCTR	PDMAC Count Register
0xD220	64	PDMCFG	PDMAC Configuration Register
0xD228	64	PDMSTATUS	PDMAC Status Register

Table 4.2.3 Internal Registers (9/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Configuration			
0xE000	64	CCFG	Chip Configuration Register
0xE008	64	REVID	Chip Revision ID Register
0xE010	64	PCFG	Pin Configuration Register
0xE018	64	TOEA	Timeout Error Access Address Register
0xE020	64	CLKCTR	Clock Control Register
0xE030	64	GARBC	G-Bus Arbiter Control Register
0xE048	64	RAMP	Register Address Mapping Register
0xE050	64	-	Reserved
Timer (Channel 0)			
0xF000	32	TMTCR0	Timer Control Register 0
0xF004	32	TMTISR0	Timer Interrupt Status Register 0
0xF008	32	TMCPRA0	Compare Address Register A 0
0xF00C	32	TMCPRB0	Compare Address Register B 0
0xF010	32	TMITMR0	Interval Timer Mode Register 0
0xF020	32	TMCCDR0	Divider Register 0
0xF030	32	TMPGMR0	Plus Generator Mode Register 0
0xF0F0	32	TMTRR0	Timer Read Register 0
Timer (Channel 1)			
0xF100	32	TMTCR1	Timer Control Register 1
0xF104	32	TMTISR1	Timer Interrupt Status Register 1
0xF108	32	TMCPRA1	Compare Address Register A 1
0xF10C	32	TMCPRB1	Compare Address Register B 1
0xF110	32	TMITMR1	Interval Timer Mode Register 1
0xF120	32	TMCCDR1	Divider Register 1
0xF130	32	TMPGMR1	Plus Generator Mode Register 1
0xF1F0	32	TMTRR1	Timer Read Register 1
Timer (Channel 2)			
0xF200	32	TMTCR2	Timer Control Register 2
0xF204	32	TMTISR2	Timer Interrupt Status Register 2
0xF208	32	TMCPRA2	Compare Register A 2
0xF210	32	TMITMR2	Interval Timer Mode Register 2
0xF220	32	TMCCDR2	Divider Register 2
0xF240	32	TMWTMR2	Watch Dog Timer Register 2
0xF2F0	32	TMTRR2	Timer Read Register 2

Table 4.2.3 Internal Registers (10/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Serial I/O (Channel 0)			
0xF300	32	SILCR0	Line Control Register 0
0xF304	32	SIDICR0	DMA/Interrupt Control Register 0
0xF308	32	SIDISR0	DMA/ Interrupt Status Register 0
0xF30C	32	SISCISR0	Status Change Interrupt Status Register 0
0xF310	32	SIFCR0	FIFO Control Register 0
0xF314	32	SIFLCR0	Flow Control Register 0
0xF318	32	SIBGR0	Baud Rate Control Register 0
0xF31C	32	SITFIFO0	Transmitter FIFO Register 0
0xF320	32	SIRFIFO0	Receiver FIFO Register 0
Serial I/O (Channel 1)			
0xF400	32	SILCR1	Line Control Register 1
0xF404	32	SIDICR1	DMA/Interrupt Control Register 1
0xF408	32	SIDISR1	DMA/ Interrupt Status Register 1
0xF40C	32	SISCISR1	Status Change Interrupt Status Register 1
0xF410	32	SIFCR1	FIFO Control Register 1
0xF414	32	SIFLCR1	Flow Control Register 1
0xF418	32	SIBGR1	Baud Rate Control Register 1
0xF41C	32	SITFIFO1	Transmitter FIFO Register 1
0xF420	32	SIRFIFO1	Receiver FIFO Register 1
Parallel I/O (PIO)			
0xF500	32	PIODO	Output Data Register
0xF504	32	PIODI	Input Data Register
0xF508	32	PIODIR	Direction Control Register
0xF50C	32	PIOOD	Open Drain Control Register

Table 4.2.3 Internal Registers (11/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Interrupt Controller (IRC)			
0xF510	32	IRFLAG0	Interrupt Request Flag 0 Register
0xF514	32	IRFLAG1	Interrupt Request Flag 1 Register
0xF518	32	IRPOL	Interrupt Request Polarity Control Register
0xF51C	32	IRRCNT	Interrupt Request Control Register
0xF520	32	IRMASKINT	Internal Interrupt Mask Register
0xF524	32	IRMASKEXT	External Interrupt Mask Register
0xF600	32	IRDEN	Interrupt Detection Enable Register
0xF604	32	IRDM0	Interrupt Detection Mode Register 0
0xF608	32	IRDM1	Interrupt Detection Mode Register 1
0xF610	32	IRLVL0	Interrupt Level Register 0
0xF614	32	IRLVL1	Interrupt Level Register 1
0xF618	32	IRLVL2	Interrupt Level Register 2
0xF61C	32	IRLVL3	Interrupt Level Register 3
0xF620	32	IRLVL4	Interrupt Level Register 4
0xF624	32	IRLVL5	Interrupt Level Register 5
0xF628	32	IRLVL6	Interrupt Level Register 6
0xF62C	32	IRLVL7	Interrupt Level Register 7
0xF640	32	IRMSK	Interrupt Mask Level Register
0xF660	32	IREDC	Interrupt Edge Detection Clear Register
0xF680	32	IRPND	Interrupt Pending Register
0xF6A0	32	IRCS	Interrupt Current Status Register

Table 4.2.3 Internal Registers (12/12)

Offset Address	Register Size (bit)	Register Symbol	Register Name
AC-link Controller (ACLC)			
0xF700	32	ACCTLEN	ACLC Control Enable Register
0xF704	32	ACCTLDIS	ACLC Control Disable Register
0xF708	32	ACREGACC	ACLC CODEC Register Access Register
0xF710	32	ACINTSTS	ACLC Interrupt Status Register
0xF714	32	ACINTMSTS	ACLC Interrupt Masked Status Register
0xF718	32	ACINTEN	ACLC Interrupt Enable Register
0xF71C	32	ACINTDIS	ACLC Interrupt Disable Register
0xF720	32	ACSEMAPH	ACLC Semaphore Register
0xF740	32	ACGPIDAT	ACLC GPI Data Register
0xF744	32	ACGPODAT	ACLC GPO Data Register
0xF748	32	ACSLTEN	ACLC Slot Enable Register
0xF74C	32	ACSLTDIS	ACLC Slot Disable Register
0xF750	32	ACFIFOSTS	ACLC FIFO Status Register
0xF780	32	ACDMASTS	ACLC DMA Request Status Register
0xF784	32	ACDMASEL	ACLC DMA Channel Selection Register
0xF7A0	32	ACAUDODAT	ACLC Audio PCM Output Data Register
0xF7A4	32	ACSURRDAT	ACLC Surround Data Register
0xF7A8	32	ACCENTDAT	ACLC Center Data register
0xF7AC	32	ACLFEDAT	ACLC LFE Data Register
0xF7B0	32	ACAUDIDAT	ACLC Audio PCM Input Data Register
0xF7B8	32	ACMODODAT	ACLC Modem Output Data Register
0xF7BC	32	ACMODIDAT	ACLC Modem Input Data Register
0xF7FC	32	ACREVID	ACLC Revision ID Register
Serial Peripheral Interface (SPI)			
0xF800	32	SPMCR	SPI Master Control Register
0xF804	32	SPCR0	SPI Control Register 0
0xF808	32	SPCR1	SPI Control Register 1
0xF80C	32	SPFS	SPI Inter Frame Space Register
0xF810	32	-	Reserved
0xF814	32	SPSR	SPI Status Register
0xF818	32	SPDR	SPI Data Register
0xF81C	32	-	Reserved

5. Configuration Registers

5.1 Detailed Description

The configuration registers set up and control the basic functionality of the entire TX4938. Refer to Section 5.2 for details of each configuration register. Also refer to sections mentioned in the description about each bit field.

5.1.1 Detecting G-Bus Timeout

The G-bus is an internal bus of the TX4938. Access to each address on the G-Bus is completed upon a bus response from the accessed address. If an attempt is made to access an undefined physical address or if a hardware failure occurs, no bus response is made. If a bus response does not occur, the bus access will not be completed, leading to a system halt. To solve this problem, the TX4938 is provided with a G-Bus timeout detection function. This function forcibly stops bus access if no bus response occurs within the specified time.

Setting the G-Bus Timeout Error Detection bit (CCFG.TOE) of the chip configuration register enables the G-Bus timeout detection function. If a bus response does not occur within the G-Bus clock (GBUSCLK) cycle specified in the G-Bus Timeout Time field (CCFG.GTOT), the G-Bus timeout detection function makes an error response to force the bus access to end. The accessed address is stored to the timeout error access address register (TOEA).

If a timeout error is detected while the TX49/H3 core, as the bus master, is gaining write access to the G-Bus, the Write-Access Bus Error bit (CCFG.BEOW) is set. Enabling interrupt No. 1 in the interrupt controller makes it possible to post an interrupt to the TX49/H3 core. If a timeout error is detected while the TX49/H3 core is gaining read access to the bus, a bus error exception occurs in the TX49/H3 core.

If a timeout error is detected while another G-Bus master (the PCI controller or DMA controller) is accessing the G-Bus, an error bit in that controller is set, which can be used to post an interrupt. Refer to the descriptions of each controller for details.

If the TRST* signal is deasserted, it is assumed that an EJTAG probe is connected, so the G-Bus timeout detection feature is disabled.

5.2 Registers

Table 5.2.1 lists the configuration registers.

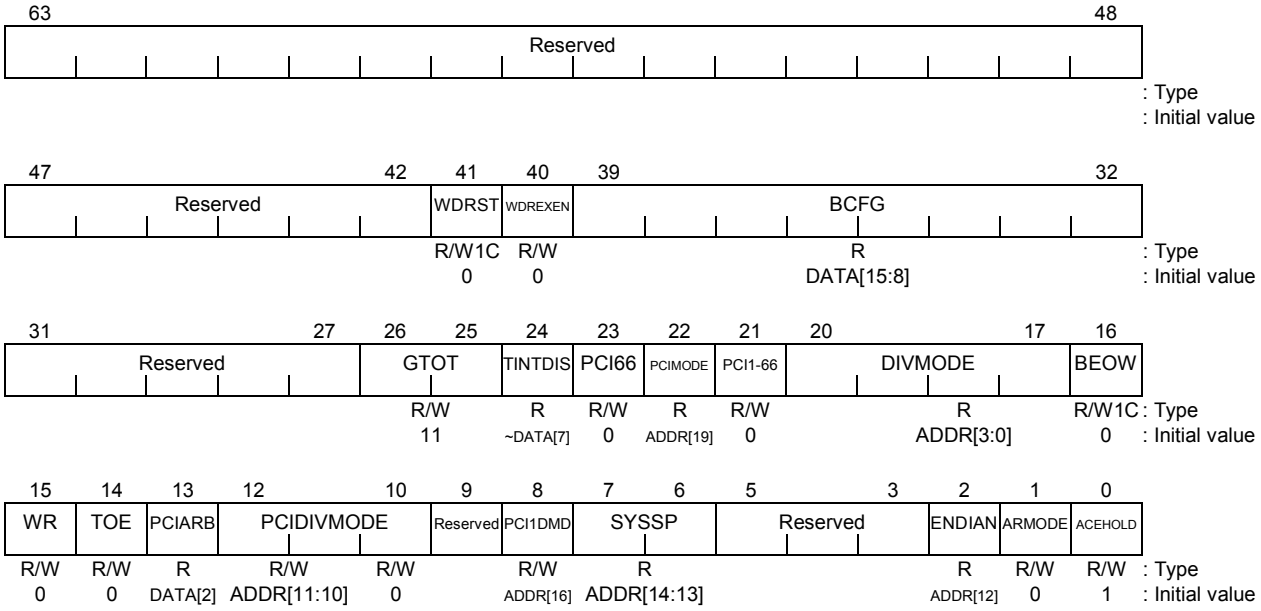
Table 5.2.1 Configuration Register Mapping

Offset Address	Size in Bits	Register Symbol	Register Name
0xE000	64	CCFG	Chip Configuration Register
0xE008	64	REVID	Chip Revision ID Register
0xE010	64	PCFG	Pin Configuration Register
0xE018	64	TOEA	Timeout Error Access Address Register
0xE020	64	CLKCTR	Clock Control Register
0xE030	64	GARBC	G-Bus Arbiter Control Register
0xE048	64	RAMP	Register Address Mapping Register
0xE058	64	JMPADR	Jump Address Register

Any address not defined in this table is reserved for future use.

5.2.1 Chip Configuration Register (CCFG) 0xE000

For the bit fields whose initial values are set by boot configuration (refer to Section 3.2), the initial input signal level and the corresponding register value are indicated.



Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:42	—	Reserved	—	—	—
41	WDRST	Watchdog Reset Status	Watch Dog Reset Status (Initial Value 0, RW1C) Indicates that a watchdog reset has occurred (refer to Section 12.3.6). Initialized when CGRESET* is asserted. 0 = No watchdog reset has occurred. 1 = A watchdog reset has occurred	0	R/W1C
40	WDREXEN	Watchdog Reset External Output	Watch Dog Reset External Enable (Initial Value 0, R/W) Specifies whether to assert the WDRST* signal at a watchdog reset (refer to Section 12.3.6). Initialized when CGRESET* is asserted. 0 = Do not assert the WDRST* signal. 1 = Assert the WDRST* signal.	0	R/W
39:32	BCFG	Boot Configuration	Set to 1 at a reset if the corresponding DATA[15:8] signal is high. Set to 0 at a reset if the corresponding DATA[15:8] signal is low.	DATA[15:8]	R
31:27	—	Reserved	—	—	—
26:25	GTOT	G-Bus Timeout Time	Specifies the number of G-Bus clock (GBUSCLK) cycles after which a bus timeout error will occur on the internal bus (G-Bus) of the TX4938. 11 = 4096 GBUSCLK 10 = 2048 GBUSCLK 01 = 1024 GBUSCLK 00 = 512 GBUSCLK	11	R/W
24	TINTDIS	Disable TX49/H3 Core Timer Interrupt	Indicates a value for indicating whether to enable the TX49/H3 internal timer interrupt (refer to Section 15.3.5). H: 0: The TX49/H3 internal timer interrupt is enabled. L: 1: The TX49/H3 internal timer interrupt is disabled.	~DATA[7]	R

Figure 5.2.1 Chip Configuration Register (1/3)

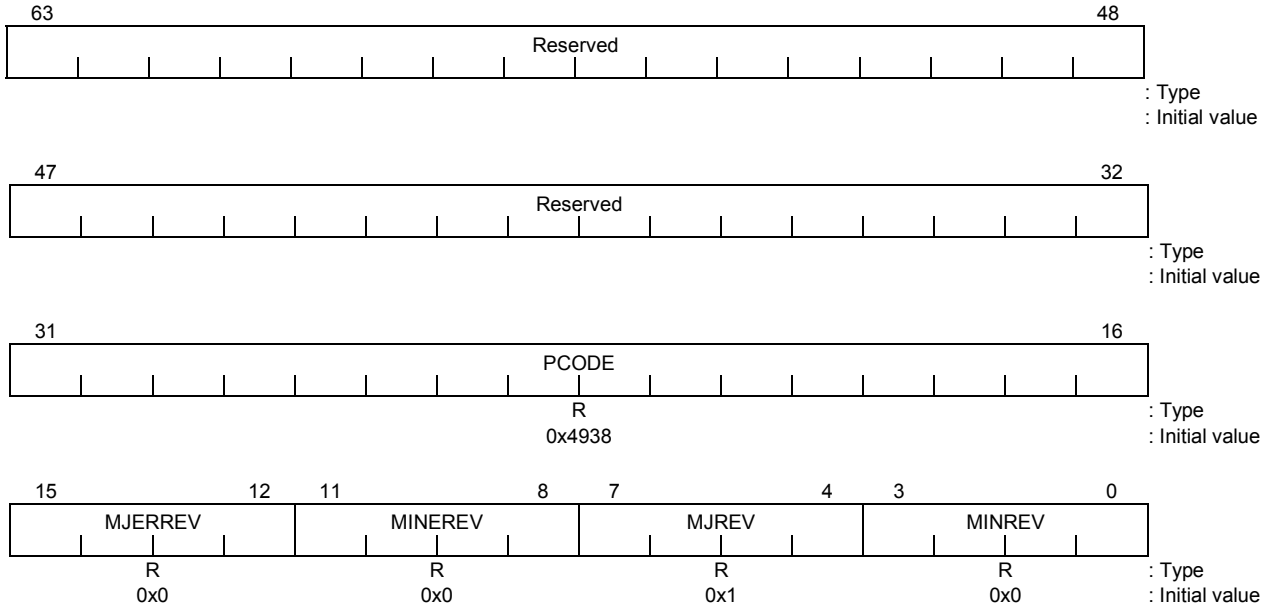
Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
23	PCI66	PCI 66MHz Mode	Used to inform the device connected to the PCI bus that a 66 MHz operation is to be performed. This bit is valid only when the PCI controller of the TX4938 is in host mode. (Refer to Section 10.3.8.) 0 = Do not perform a 66 MHz operation. 1 = Perform a 66 MHz operation.	0	R/W
22	PCIMODE	PCI Operation Mode	Indicates information about the operation mode of the TX4938 PCI controller. (Refer to Section 10.3.1.) L: 0: Satellite mode H: 1: Host mode	ADDR[19]	R
21	PCI1-66	PCI1 66MHz Mode	Used to inform the ETHERC0 and ETHERC1 connected to the PCI1 bus that a 66 MHz operation is to be performed. 0 = Do not perform a 66 MHz operation (1/4 GBUSCLK). 1 = Perform a 66 MHz operation (1/2 GBUSCLK).	0	R/W
20:17	DIVMODE	CPUCLK Frequency Multiplication Factor	Indicates information about the frequency multiplication factor of the TX49/H3 core clock (CPUCLK) to the MASTERCLK. This field is set with a result of encoding an initial input value at ADDR[3:0]. The PLL incorporated in the TX4938 multiplies the MASTERCLK and supplies the resulting frequency to the TX49/H3 core. The value set in DIVMODE[3:0] is reflected in the EC field of the TX49/H3 core Config register. ADDR[3:0]:DIVMODE[3:0] HHHH: 0100: CPUCLK freq. = 2 × MASTERCLK freq. HHHL: 1111: CPUCLK freq. = 2.5 × MASTERCLK freq. HHLH: 0101: CPUCLK freq. = 3 × MASTERCLK freq. HHLL: 0110: CPUCLK freq. = 4 × MASTERCLK freq. LHHH: 1101: CPUCLK freq. = 4.5 x MASTERCLK freq. LHHL: ---- : Reserved LHLH: ---- : Reserved LHLL: ---- : Reserved HLHH: 0000: CPUCLK freq. = 8 × MASTERCLK freq. HLHL: 1011: CPUCLK freq. = 10 × MASTERCLK freq. HLLH: 0001: CPUCLK freq. = 12 × MASTERCLK freq. HLLL: 0010: CPUCLK freq. = 16 × MASTERCLK freq. LLHH: 1001: CPUCLK freq. = 18 × MASTERCLK freq. LLHL: ---- : Reserved LLLH: ---- : Reserved LLLL: ---- : Reserved	ADDR[3:0]	R
16	BEOW	Write-Access Bus Error	Indicates that a timeout error has occurred in the internal bus (G-Bus) during a write bus transaction of the TX49/H3 core. This bit corresponds to interrupt No. 1 in the interrupt controller. 0 = No error has occurred. 1 = An error has occurred.	0	R/W1C
15	WR	Watchdog Timer Mode	Specifies how information will be reported in watchdog timer mode (refer to Section 12.3.6). 0 = Generate an NMI exception. 1 = Generate a watchdog reset.	0	R/W
14	TOE	G-Bus Timeout Error Detection	Specifies whether to detect and report a bus timeout error in the internal bus (G-Bus) of the TX4938. 0 = Do not detect or report a bus timeout error. 1 = Detect and report a bus timeout error.	0	R/W
13	PCIARB	PCI Arbiter Selection	Indicates the PCI bus arbiter selection setting (refer to Section 10.3.12). L: 0 = External PCI bus arbiter H: 1 = Built-in PCI bus arbiter	DATA[2]	R

Figure 5.2.1 Chip Configuration Register (2/3)

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
12:10	PCIDIVMODE	PCICLK Frequency Division Ratio	Specifies the frequency division ratio of the PCI bus clock output (PCICLK[5:0]) frequency to the clock frequency (CPUCLK) of the TX49/H3 core. 001: PCICLK frequency = CPUCLK frequency ÷ 4 011: PCICLK frequency = CPUCLK frequency ÷ 4.5 101: PCICLK frequency = CPUCLK frequency ÷ 5 111: PCICLK frequency = CPUCLK frequency ÷ 5.5 000: PCICLK frequency = CPUCLK frequency ÷ 8 010: PCICLK frequency = CPUCLK frequency ÷ 9 100: PCICLK frequency = CPUCLK frequency ÷ 10 110: PCICLK frequency = CPUCLK frequency ÷ 11	ADDR[11:10] ,0	R/W
9	—	Reserved	—	—	—
8	PCI1DMD	PCI1CLK Frequency Division Ratio	Internal PCI1CLK Frequency Division Ration L 0: PCI1CLK frequency = GBUSCLK frequency ÷ 2 H 1: PCI1CLK frequency = GBUSCLK frequency ÷ 4	ADDR[16]	R
7:6	SYSSP	SYSCLK frequency division ratio	Indicates the frequency division ratio of the SYSCLK frequency to the G-Bus clock frequency (GBUSCLK). LL: 00: SYSCLK frequency = GBUSCLK frequency ÷ 4 LH: 01: SYSCLK frequency = GBUSCLK frequency ÷ 3 HL: 10: SYSCLK frequency = GBUSCLK frequency ÷ 2 HH: 11: SYSCLK frequency = GBUSCLK frequency	ADDR[14:13]	R
5:3	—	Reserved	—	—	—
2	ENDIAN	Endian	Indicates the TX4938 endian mode setting. L: 0 = Little endian mode H: 1 = Big endian mode	ADDR[12]	R
1	ARMODE	ACK*/READY Mode	Selects an ACK*/READY signal operation mode for the external bus controller (refer to Section 7.3.6). 0 = ACK*/READY dynamic mode 1 = ACK*/READY static mode	0	R/W
0	ACEHOLD	ACE Hold	Specifies the hold time of an address relative to the external bus controller ACE* signal (refer to Section 7.3.4). 0 = Switch the address at the same time when the ACE* signal is deasserted. 1 = Switch the address one clock cycle after the ACE* signal is deasserted.	1	R/W

Figure 5.2.1 Chip Configuration Register (3/3)

5.2.2 Chip Revision ID Register (REVID) 0xE008

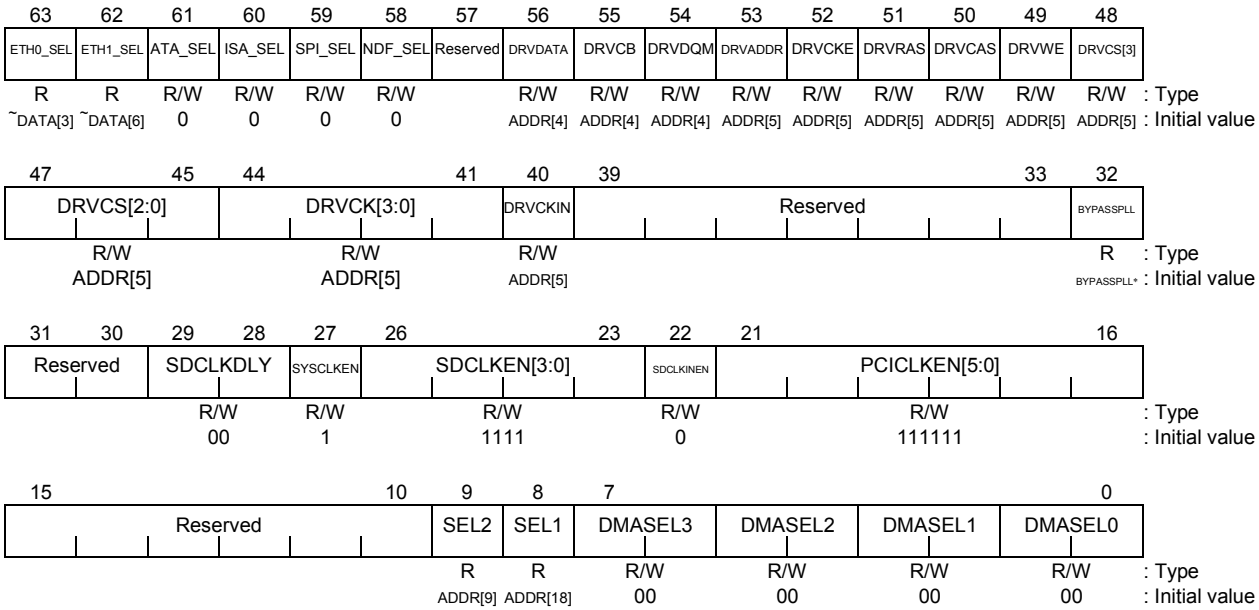


Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:32	—	Reserved	—	—	—
31:16	PCODE	Product Code	Indicates the product number. It is a fixed value.	0x4938	R
15:12	MJERREV	Major Extra Code	Indicates the major extra code.	0x0	R
11:8	MINEREV	Major Extra Code	Indicates the minor extra code.	0x0	R
7:4	MJREV	Major Revision Code	Indicates the major revision of the product. Contact Toshiba technical staff for the latest information.	0x1	R
3:0	MINREV	Minor Revision Code	Indicates the minor revision of the product. Contact Toshiba technical staff for the latest information.	0x0	R

Figure 5.2.2 Chip Revision ID Register

5.2.3 Pin Configuration Register (PCFG) 0xE010

For the bit fields whose initial values are set by boot configuration (refer to Section 3.2), the initial input signal level and the corresponding register value are indicated.


63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48
ETH0_SEL ETH1_SEL ATA_SEL ISA_SEL SPI_SEL NDF_SEL Reserved DRVDATA DRVCB DRVDQM DRVADDR DRVCKE DRVRAS DRVCAS DRVWE DRVCS[3]
R R R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W : Type
~DATA[3] ~DATA[6] 0 0 0 0 ADDR[4] ADDR[4] ADDR[4] ADDR[5] ADDR[5] ADDR[5] ADDR[5] ADDR[5] ADDR[5] : Initial value
47 45 44 41 40 39 33 32
DRVCS[2:0] DRVCK[3:0] DRVCKIN Reserved BYPASSPLL
R/W R/W R/W R : Type
ADDR[5] ADDR[5] ADDR[5] BYPASSPLL* : Initial value
31 30 29 28 27 26 23 22 21 16
Reserved SDCLKDLY SYSCLKEN SDCLKEN[3:0] SDCLKINEN PCICLKEN[5:0]
R/W R/W R/W R/W R/W : Type
00 1 1111 0 111111 : Initial value
15 10 9 8 7 0
Reserved SEL2 SEL1 DMASEL3 DMASEL2 DMASEL1 DMASEL0
R R R/W R/W R/W R/W : Type
ADDR[9] ADDR[18] 00 00 00 00 : Initial value

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63	ETH0_SEL	Status of shared pin	ETHERC0 Shared-pin status Indicates the status of shared pin. L : 1 : ETHERC0 selected H : 0 : ETHERC0 not selected	~DATA[3]	R
62	ETH1_SEL	Status of shared pin	ETHERC1 Shared-pin status Indicates the status of shared pin. L : 1 : ETHERC1 selected H : 0 : ETHERC1 not selected	~DATA[6]	R
61	ATA_SEL	Setting of shared pin	ATA/ISA Shared-pin setting Specifies the status of shared pin. 1 : ATA/ISA selected 0 : ATA/ISA not selected	0	R/W
60	ISA_SEL	Setting of shared pin	ISA Shared-pin setting Specifies the status of shared pin. 1 : ISA selected 0 : ISA not selected	0	R/W
59	SPI_SEL	Setting of shared pin	SPI Shared-pin setting Specifies the status of shared pin. 1 : SPI selected 0 : SPI not selected	0	R/W
58	NDF_SEL	Setting of shared pin	NDFMC Shared-pin setting Specifies the status of shared pin. When use NAND Flash IPL function, the initial value of this bit indicates 1 after booting. 1 : NDFMC selected 0 : NDFMC not selected	0	R/W
57	—	Reserved	—	—	—

Figure 5.2.3 Pin Configuration Register (1/3)

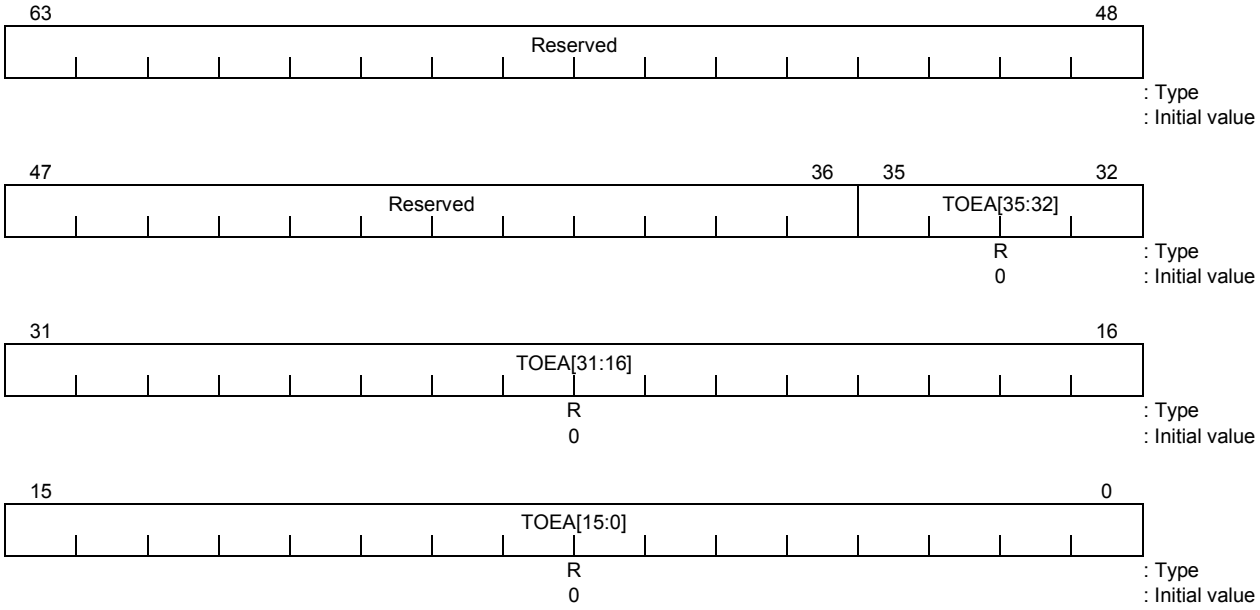
Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
56	DRVDATA	DATA Signal Control	Specifies the driving capability of the DATA[63:0] signals. L : 0 = 8 mA H : 1 = 16 mA	ADDR[4]	R/W
55	DRVCB	CB Signal Control	Specifies the driving capability of the CB[7:0]* signals. L : 0 = 8 mA H : 1 = 16 mA Note: CB[7:0]* share pins with PIO[15:8], E0TXD[3:0], E0RXD[3:0]. The driving capability of these pins are below. CB[7:0], E0TXD[3:0], E0RXD[3:0]: 8 mA or 16 mA PIO[15:8]: 8 mA only	ADDR[4]	R/W
54	DRVDQM	DQM Signal Control	Specifies the driving capability of the DQM[7:0]* signals. L : 0 = 8 mA H : 1 = 16 mA	ADDR[4]	R/W
53	DRVADDR	ADDR Signal Control	Specifies the driving capability of the ADDR[19:0] signals. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
52	DRVCKE	CKE Signal Control	Specifies the driving capability of the CKE signal. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
51	DRVRAS	RAS Signal Control	Specifies the driving capability of the RAS* signal. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
50	DRVCAS	CAS Signal Control	Specifies the driving capability of the CAS* signal. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
49	DRVWE	WE Signal Control	Specifies the driving capability of the WE* signal. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
48:45	DRVCS[3:0]	SDRAM CS Signal Control	Specifies the driving capability of the SDCS[3:0]* signals. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
44:41	DRVCK[3:0]	SDRAM SDCLK Signal Control	Specifies the driving capability of the SDCLK[3:0] signals. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
40	DRVCKIN	SDRAM SDCLKIN Signal Control	Specifies the driving capability of the SDCLKIN signal. L : 0 = 8 mA H : 1 = 16 mA	ADDR[5]	R/W
39:33	—	Reserved	—	—	—
32	BYPASS PLL	Bypass PLL	Indicates information about whether a PLL for a circuit other than the PCI controller is on or off. L: 0 = The PLL is off. H: 1 = The PLL is on.	BYPASSPLL*	R
31:30	—	Reserved	—	—	—
29:28	SDCLKDLY	SDCLK Feedback Delay	Specifies the feedback delay for the SDCLK. This function is for diagnosis purposes. Usually, set the bits to 00. 00 = Delay 1 (minimum delay) 10 = Delay 2 01 = Delay 3 11 = Delay 4 (maximum delay)	00	R/W
27	SYSCLKEN	SYSCLK Enable	Specifies whether to output the SYSCLK. 1 = Clock output 0 = H	1	R/W

Figure 5.2.3 Pin Configuration Register (2/3)

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
26:23	SDCLKEN [3:0]	SDCLK Enable	Individually specifies whether to output each of SDCLK[3:0]. 1 = Clock output 0 = H Bit 26 = SDCLK[3] Bit 25 = SDCLK[2] Bit 24 = SDCLK[1] Bit 23 = SDCLK[0]	1111	R/W
22	SDCLKINEN	SDCLKIN Enable	Specifies how SDCLK[3:0] should be fed back. This function is for diagnosis purposes. Usually, set this bit to 0. 0 = Use the SDCLKIN signal as a feedback clock. 1 = Perform feedback within the TX4938 (the SDCLKIN becomes an output signal).	0	R/W
21:16	PCICLKEN [5:0]	PCICLK Enable	Individually specifies whether to output each of PCICLK[5:0]. 1 = Clock output 0 = H Bit 21 = PCICLK[5] Bit 20 = PCICLK[4] Bit 19 = PCICLK[3] Bit 18 = PCICLK[2] Bit 17 = PCICLK[1] Bit 16 = PCICLK[0]	111111	R/W
15:10	—	Reserved	—	—	—
9	SEL2	Shared-Pin Status 2	DMAREQ[2], DMAACK[2], and PIO[4:2] share pins with the AC-link interface signals. Indicates which function the shared pins are set to. L: 0 = The shared pins are set to DMAREQ[2], DMAACK[2], and PIO[4:2]. H: 1 = The shared pins are set to the AC-link interface signals.	ADDR[9]	R
8	SEL1	Shared-Pin Status 1	Indicates which function, PIO[15:8] or CB[7:0], the shared pins are set to. L: 0 = The shared pins are set to PIO[15:8]. H: 1 = The shared pins are set to CB[7:0].	ADDR[18]	R
7:4	—	Reserved	—	—	—
3	DMASEL3	DMA Request Select 3	Selects a DMA request used by DMA controller 0 channel 3. 0: DMAREQ[3] (external) 1: SIO channel 0 transmission (internal)	0	R/W
2	DMASEL2	DMA Request Select 2	Selects a DMA request used by DMA controller 0 channel 2. 0: DMAREQ[2] (external) 1: SIO channel 0 reception (internal)	0	R/W
1	DMASEL1	DMA Request Select 1	Selects a DMA request used by DMA controller 0 channel 1. 00: DMAREQ[1] (external) 01: SIO channel 1 transmission (internal)	0	R/W
1:0	DMASEL0	DMA Request Select 0	Selects a DMA request used by DMA controller 0 channel 0. 00: DMAREQ[0] (external) 01: SIO channel 1 reception (internal)	0	R/W

Figure 5.2.3 Pin Configuration Register (3/3)

5.2.4 Timeout Error Access Address Register (TOEA) 0xE018

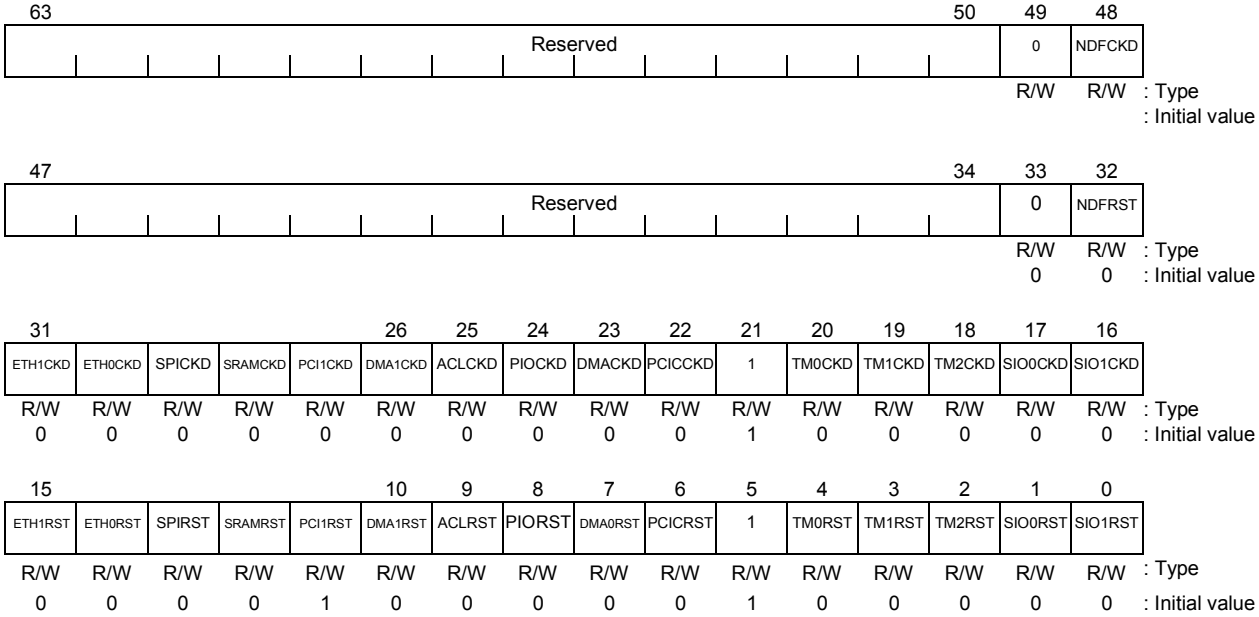


Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:36	—	Reserved	—	—	—
35:0	TOEA	Timeout Error Access Address	Holds the G-Bus address for the G-Bus cycle in which the latest G-Bus timeout error was detected.	0x0_0000_0000	R

Figure 5.2.4 Timeout Error Access Address Register

5.2.5 Clock Control Register (CLKCTR) 0xE020

Bit 32 and bits 15-0 are reset bits for the on-chip peripheral modules. To bring on-chip peripheral modules out of the reset state, the corresponding bits must be cleared by software. Before clearing them, wait at least 128 CPU clock cycles after they are set.


63 50 49 48
Reserved
0
NDFCKD
R/W R/W : Type
: Initial value
47 34 33 32
Reserved
0
NDFRST
R/W R/W : Type
0 0 : Initial value
31 26 25 24 23 22 21 20 19 18 17 16
ETH1CKD ETH0CKD SPICKD SRAMCKD PCI1CKD DMA1CKD ACLCKD PIOCKD DMACKD PCICCKD 1 TM0CKD TM1CKD TM2CKD SIO0CKD SIO1CKD
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W : Type
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 : Initial value
15 10 9 8 7 6 5 4 3 2 1 0
ETH1RST ETH0RST SPIRST SRAMRST PCI1RST DMA1RST ACLRST PIORST DMA0RST PCICRST 1 TM0RST TM1RST TM2RST SIO0RST SIO1RST
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W : Type
0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 : Initial value

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:50	—	Reserved	—	—	—
49	—	—	Always set this bit to 0.	0	R/W
48	NDFCKD	NDFMC Clock Disable	Controls clock pulses for the NAND Flash Memory controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
47:34	—	Reserved	—	—	—
33	—	—	Always set this bit to 0.	0	R/W
32	NDFRST	NDFMC Reset	Resets the NAND Flash Memory controller. 0 = Normal state 1 = Reset	0	R/W
31	ETH1CKD	ETHERC1 Clock Disable	Controls clock pulses for the ETHERC1 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
30	ETH0CKD	ETHERC0 Clock Disable	Controls clock pulses for the ETHERC0 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
29	SPICKD	SPIC Clock Disable	Controls clock pulses for the SPI controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
28	SRAMCKD	SRAMC Clock Disable	Controls clock pulses for the internal SRAM controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
27	PCIC1CKD	PCIC1 Clock Disable	Controls clock pulses for PCI controller 1. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
26	DMA1CKD	DMAC1 Clock Disable	Controls clock pulses for the DMA controller 1. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W

Figure 5.2.5 Clock Control Register (1/3)

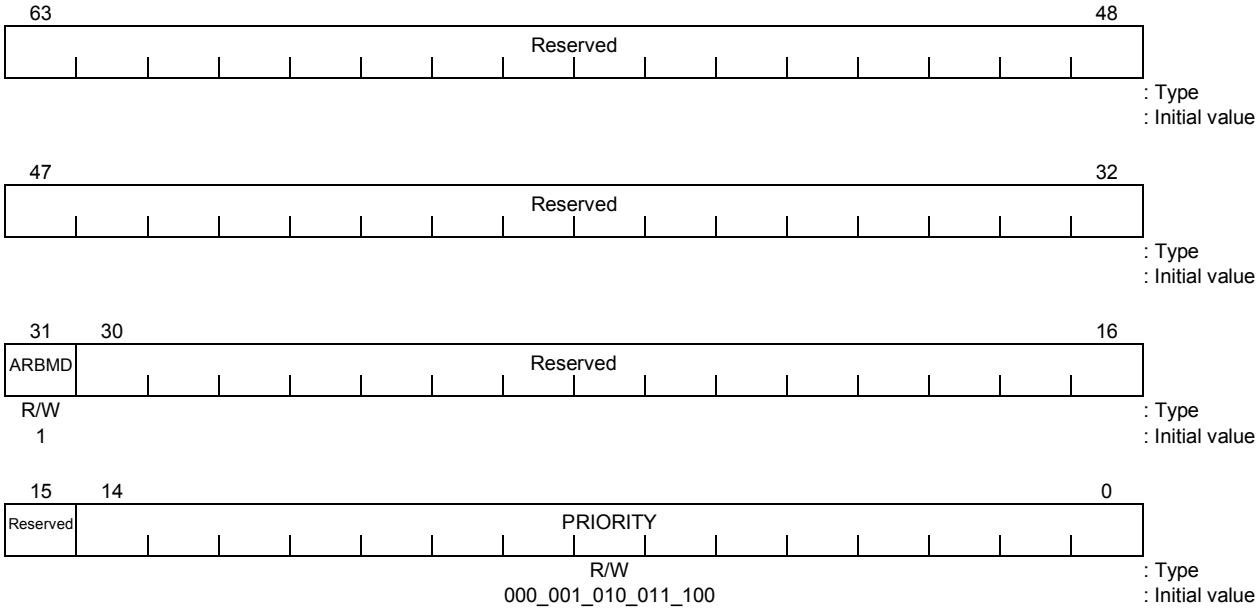
Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
25	ACLCKD	ACLC Clock Disable	Controls clock pulses for the AC-link controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
24	PIOCKD	PIO Clock Disable	Controls clock pulses for the parallel IO controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
23	DMA0CKD	DMAC0 Clock Disable	Controls clock pulses for the DMA controller 0. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
22	PCICKD	PCIC Clock Disable	Controls clock pulses for the PCI controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
21	—	—	Always set this bit to 1.	1	R/W
20	TM0CKD	Timer 0 Clock Disable	Controls clock pulses for the TMR0 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
19	TM1CKD	Timer 1 Clock Disable	Controls clock pulses for the TMR1 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
18	TM2CKD	Timer 2 Clock Disable	Controls clock pulses for the TMR2 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
17	SIO0CKD	SIO0 Clock Disable	Controls clock pulses for the SIO0 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
16	SIO1CKD	SIO1 Clock Disable	Controls clock pulses for the SIO1 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
15	ETHC1RST	ETHERC1 Reset	Resets the ETHER controller 1. 0 = Normal state 1 = Reset	0	R/W
14	ETHC0RST	ETHERC0 Reset	Resets the ETHER controller 0. 0 = Normal state 1 = Reset	0	R/W
13	SPIRST	SPI Reset	Resets the SPI controller. 0 = Normal state 1 = Reset	0	R/W
12	SRAMRST	SRAMC Reset	Resets the internal SRAM controller. 0 = Normal state 1 = Reset	0	R/W
11	PCIC1RST	PCIC1 Reset	Resets the PCI controller 1. 0 = Normal state 1 = Reset	1	R/W
10	DMA1RST	DMAC1 Reset	Resets the DMAC controller 1. 0 = Normal state 1 = Reset	0	R/W
9	ACLRST	ACLC Reset	Resets the AC-link controller. 0 = Normal state 1 = Reset Note: Reset the AC-link controller when it is not asserting the interrupt and DMA request.	0	R/W
8	PIORST	PIO Reset	Resets the parallel IO controller. 0 = Normal state 1 = Reset	0	R/W

Figure 5.2.5 Clock Control Register (2/3)

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
7	DMARST	DMAC Reset	Resets the DMA controller. 0 = Normal state 1 = Reset	0	R/W
6	PCICRST	PCIC Reset	Resets the PCI controller. 0 = Normal state 1 = Reset	0	R/W
5	—	—	Always set this bit to 1.	1	R/W
4	TM0RST	TMR0 Reset	Resets the TMR0 controller. 0 = Normal state 1 = Reset	0	R/W
3	TM1RST	TMR1 Reset	Resets the TMR1 controller. 0 = Normal state 1 = Reset	0	R/W
2	TM2RST	TMR2 Reset	Resets the TMR2 controller. 0 = Normal state 1 = Reset	0	R/W
1	SIO0RST	SIO0 Reset	Resets the SIO0 controller. 0 = Normal state 1 = Reset	0	R/W
0	SIO1RST	SIO1 Reset	Resets the SIO1 controller. 0 = Normal state 1 = Reset	0	R/W

Figure 5.2.5 Clock Control Register (3/3)

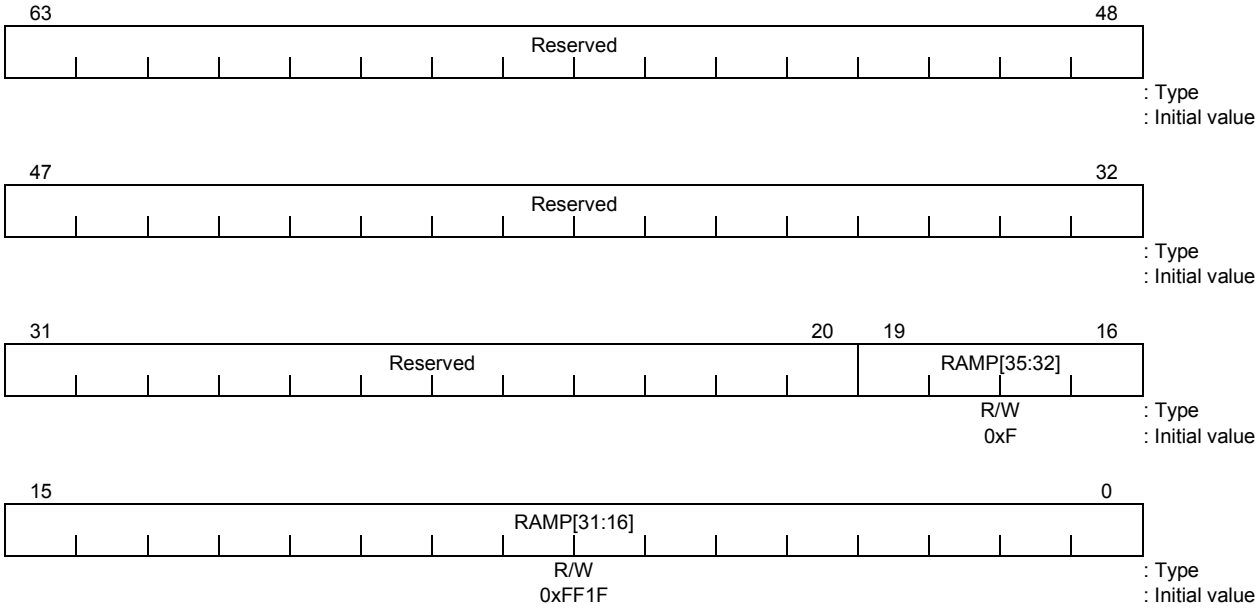
5.2.6 G-Bus Arbiter Control Register (GARBC) 0xE030



Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:32	—	Reserved	—	—	—
31	ARBMD	Arbitration Mode	Specifies how to prioritize G-Bus arbitration. 0 = Fixed priority. The G-Bus arbitration priority conforms to the content of the PRIORITY field (bits [14:0]). 1 = Round-robin (in a round-robin fashion, PCIC0 > PDMAC > DMAC0 > DMAC1 > PCIC1) Note: Before accessing the PCI by DMAC, specify round-robin as the priority mode. If fixed-priority mode is selected, a dead lock is likely to occur in PCI bus access.	1	R/W
30:15	—	Reserved	—	—	—
14:0	PRIORITY	Arbitration Priority	Specifies the priority when ARBMD (bit [16]) specifies fixed-priority mode. [14:12] = Bus master with the highest priority [11:9] = Bus master with the second highest priority [8:6] = Bus master with the third highest priority [5:3] = Bus master with the fourth highest priority [2:0] = Bus master with the fifth highest priority 000 = PCI controller 001 = PDMAC 010 = DMAC0 011 = DMAC1 100 = PCI controller 1 A priority of PCIC > PDMAC > DMAC0 > DMAC1 > PCIC1 is initially set up.	000_001_010_011_100	R/W

Figure 5.2.6 G-Bus Arbiter Control Register

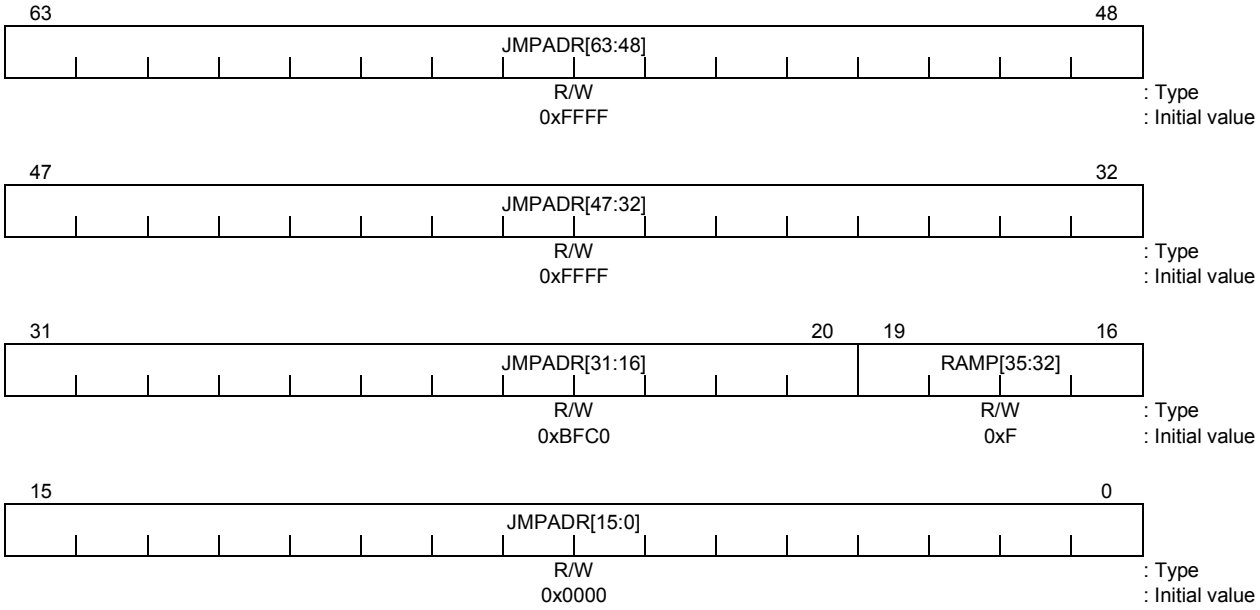
5.2.7 Register Address Mapping Register (RAMP) 0xE048



Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:20	—	Reserved	—	—	—
19:0	RAMP[35:16]	Register Address Mapping	This is a base address register for the TX4938 built-in registers. It holds the high-order 20 bits of a register address. The default built-in register base address is 0xF_FF1F_0000. Even after the content of the base address register is changed, the default value can be used to reference the built-in registers. (Refer to "4.2 Register Mapp".)	0xF_FF1F	R/W

Figure 5.2.7 Register Address Mapping Register

5.2.8 Jump Address Register (JMPADR) 0xE058



Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:0	JMPADR[63:0]	Jump Address Register	When NAND-IPL loads a user boot program to RAM from a NAND flash memory, the address of the loading place is set as this register. It is used in case NAND-IPL controls an interrupt vector table henceforth. This register is a register used on a program of NAND-IPL, and a user must not operate it directly. At the time of operating it cannot be guaranteed of operation. Please Refer to “Chapter 18 NAND Flash Memory IPL” for Details.	0xFFFF_FFFF _BFC0_0000	R/W

Figure 5.2.8 Jump Address Register

6. Clocks

6.1 TX4938 Clock Signals

Figure 6.1.1 shows the configuration of TX4938 blocks and clock signals. Table 6.1.1 describes each clock signal. Table 6.1.2 shows the relationship among different clock signals when the CPU clock frequency is 266 MHz. Table 6.1.3 shows the relationship among different clock signals when the CPU clock frequency is 300 MHz. Table 6.1.4 shows the relationship among different clock signals when the CPU clock frequency is 333 MHz.

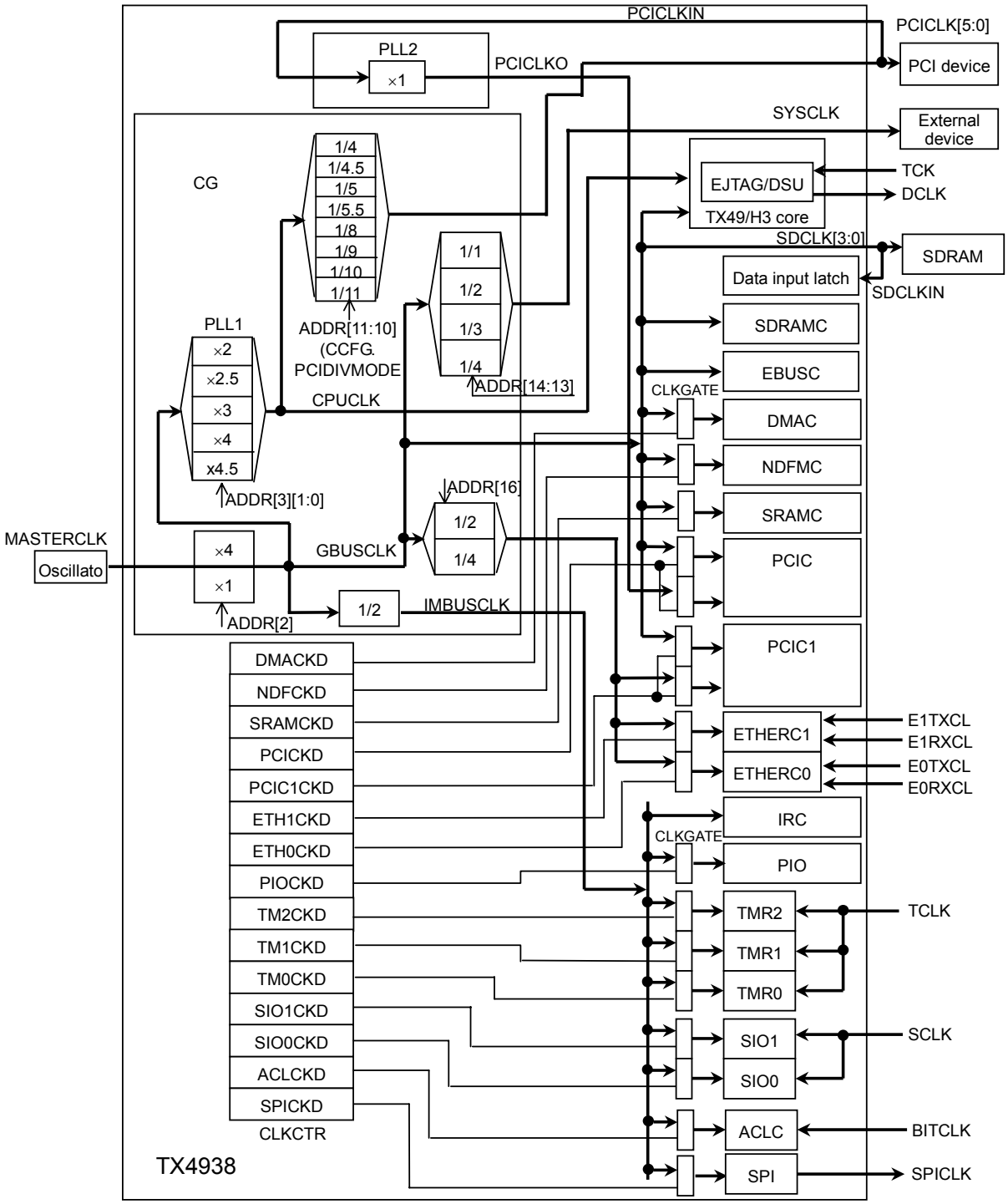


Figure 6.1.1 TX4938 Block and Clock Configuration

Table 6.1.1 TX4938 Clock Signals (1/2)

Clock	Input/Output	Description	Related Configuration Signals (Refer to Section 3.2.)	Related Registers (Refer to Chapters 5 and 10.)
MASTERCLK	Input	Master input clock for the TX4938. The TX4938 internal clock generator multiplies or divides MASTERCLK to generate internal clock pulses.	—	—
CPUCLK	Internal signal	Clock supplied to the TX49/H3 core. PLL1 in the TX4938 generates CPUCLK by multiplying MASTERCLK. Boot configuration signals ADDR[3:0] can set the frequency ratio of CPUCLK to MASTERCLK. ADDR[3:0] HHHH = 2 times MASTERCLK HHHL = 2.5 times MASTERCLK HHLH = 3 times MASTERCLK HHLL = 4 times MASTERCLK LHHH = 4.5 times MASTERCLK HLHH = 8 times MASTERCLK HLHL = 10 times MASTERCLK HLLH = 12 times MASTERCLK HLLL = 16 times MASTERCLK LLHH = 18 times MASTERCLK	ADDR[3:0]	CCFG.DIVMODE[3:0]
GBUSCLK	Internal signal	Clock supplied to peripheral blocks on the G-Bus. PLL1 in the TX4938 generates GBUSCLK by multiplying MASTERCLK. Boot configuration signal ADDR[2] can set the multiplier value. ADDR[2] L = 4 times MASTERCLK H = 1 times MASTERCLK	ADDR[2]	CCFG.DIVMODE[2]
IMBUSCLK	Internal signal	Clock supplied to peripheral modules on the IM-Bus. The frequency of IMBUSCLK is half that of GBUSCLK.	—	—
SYSCLK	Output	System clock output from the TX4938. Used by the devices connected to the external bus controller (EBUSC). Boot configuration signals ADDR[14:13] can set the frequency ratio of SYSCLK to GBUSCLK. ADDR[14:13] LL: GBUSCLK divided by 4 LH: GBUSCLK divided by 3 HL: GBUSCLK divided by 2 HH: GBUSCLK divided by 1 The SYSCLKEN bit of the PCFG register can disable the output of SYSCLK. Note: To use SYSCLK to access external devices, the SYSCLK rate must match the EBUSC channel operating rate. For details, refer to Section 7.3.8.	ADDR[14:13]	CCFG.SYSSP PCFG.SYSCLKEN
SDCLK[3:0]	Output	Clock supplied to SDRAM. The frequency of SDCLK[3:0] is the same as that of GBUSCLK. The SDCLKEN[3:0] field of the PCFG register can disable the output of SDCLK[3:0] on a per bit basis.	—	PCFG.DRVCK[3:0] PCFG.SDCLKEN[3:0]
SDCLKIN	Input/output	Reference clock used to latch input data signals from SDRAM. The clock output from SDCLK should be connected to SDCLKIN via a feedback line outside the TX4938.	—	PCFG.DRVCKIN (PCFG.SDCLKDLY) (PCFG.SDCLKINEN)

Table 6.1.1 TX4938 Clock Signals (2/2)

Clock	Input/Output	Description	Related Configuration Signals (Refer to Section 3.2.)	Related Registers (Refer to Chapters 5 and 10.)
PCICLK[5:0]	Output	Clock supplied to devices on the PCI bus. The PCICLKEN bit of the PCFG register can disable the output of PCICLK. The frequency depends on boot configuration signals ADDR[11:10] or the PCIDIVMODE field of the CCFG register. Initial Value of PCIDIVMODE[0] is 0. CCFG_PCIDIVMODE[2:0] =001: CPUCLK divided by 4 =011: CPUCLK divided by 4.5 =101: CPUCLK divided by 5 =111: CPUCLK divided by 5.5 =000: CPUCLK divided by 8 =010: CPUCLK divided by 9 =100: CPUCLK divided by 10 =110: CPUCLK divided by 11 Note: PCICLK[5:0] can supply clock pulses at 66 or 33 MHz when the CPUCLK frequency is set to 300. The setting is: 011, 010	ADDR[11:10]	CCFG.PCIDIVMODE PCFG.PCICLKEN[5:0]
PCICLKIN	Input	PCI bus clock. The built-in PCI controller of the TX4938 operates with this clock. Note: To achieve an accurate phase match with the external clock, PCICLK[5:0] or the PCI clock output from another PCI device must be supplied to PCICLKIN.	—	—
PCICLKO	Internal signal	Clock supplied to the PCI controller. PCICLKO is generated by PLL2 based on PCICLKIN. PCICLKO has the same frequency and phase as those of PCICLKIN (input pin).	—	—
EEPROM_SK	Output	Clock for serial EEPROM used to initially set the PIC configuration.	—	—
SCLK	Input	Input clock for SIO. SCLK is shared by SIO0 and SIO1.	—	—
TCLK	Input	Input clock for timers. TCLK is shared by TMR0, TMR1, and TMR2.	—	—
BITCLK	Input	Input clock for the AC-link controller. The pin is shared with the PIO[2] signal.	ADDR[9]	—
TCK	Input	Input clock for JTAG.	—	—
DCLK	Output	Clock output for the real-time debugging system.	—	—
E0TXCLK, E1TXCLK	Input	Transmit clock		
E0RXCLK, E1RXCLK	Input	Receive clock		
SPICLK	Output	This clock is used for a data clock to or from an SPI slave device.		
PCI1CLK	Internal signal	Clock supplied to the PCI controller connected ETHER0 and ETHER1. This clock is generated by dividing GBUSCLK. Divide mode is specified by ADDR[16] and CCFG.	ADDR[16]	CCFG.PCI1DMD

Table 6.1.2 Relationship Among Different Clock Frequencies (for TMPR4938XBG-300, CPUCLK = 266 MHz)

Master Clock (Input) and Boot Configured Settings		Internal Clock			External Clock (Output)												
						SYSCLK (MHz)				PCICLK[5:0] (MHz) †							
						Boot Configured Settings PCIDIVMODE[2:0]				CCFG Settings PCIDIVMODE[2:0]							
MASTERCLK (MHz)	Boot Configured Setting ADDR[3:0]	CPUCLK (MHz)	GBUSCLK (MHz)	IMBUSCLK (MHz)	SDCLK [3:0] (MHz)	HH (1/1)	HL (1/2)	LH (1/3)	LL (1/4)	LLH (1/4)	LHH (1/4.5)	HLH (1/5)	HHH (1/5.5)	LLL (1/8)	LHL (1/9)	HLL (1/10)	HHL (1/11)
133	HHHH (×2.0)	266	133	66	133	133	66	44.3	33.3	66.5	59.1	53.2	48.4	33.3	29.6	26.6	24.2
33.3	HLHH (×8.0)																
106.4	HHHL (×2.5)		106.4	53	106.4	106.4	53	35.5	26.6								
26.6	HLHL (×10.0)																
88.7	HHLH (×3.0)		88.7	44	88.7	88.7	44.3	29.6	22.2								
22.2	HLLH (×12.0)																
66.5	HHLL (×4.0)		66.5	33.2	66.5	66.5	33.3	22.2	16.6								
16.6	HLLL (×16.0)																
59.1	LHHH (×4.5)		59.1	29.6	59.1	59.1	29.6	19.7	14.8								
14.8	LLHH (×18.0)																

† The CCFG.PCIDIVMODE[2:1] field is setting by the boot configuration ADDR[11:10].

Table 6.1.3 Relationship Among Different Clock Frequencies (for TMPR4938XBG-300, CPUCLK = 300 MHz)

Master Clock (Input) and Boot Configured Settings		Internal Clock			External Clock (Output)												
						SYSCLK (MHz)				PCICLK[5:0] (MHz) †							
						Boot Configured Settings PCIDIVMODE[2:0]				CCFG Settings PCIDIVMODE[2:0]							
MASTERCLK (MHz)	Boot Configured Setting ADDR[3:0]	CPUCLK (MHz)	GBUSCLK (MHz)	IMBUSCLK (MHz)	SDCLK [3:0] (MHz)	HH (1/1)	HL (1/2)	LH (1/3)	LL (1/4)	LLH (1/4)	LHH (1/4.5)	HLH (1/5)	HHH (1/5.5)	LLL (1/8)	LHL (1/9)	HLL (1/10)	HHL (1/11)
-	HHHH(x2.0)	300	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	HLHH(x8.0)																
120	HHHL (x2.5)		120	60	120	120	60	40	30	75	66	60	54	37	33	30	27
30	HLHL (x10.0)																
100	HHLH (x3.0)		100	50	100	100	50	33	25								
25	HLLH (x12.0)																
75	HHLL (x4.0)		75	37	75	75	37	25	18								
18	HLLL (x16.0)																
66	LHHH(x4.5)		66	33	66	66	33	22	16								
16	LLHH (x18.0)																

† The CCFG.PCIDIVMODE[2:1] field is setting by the boot configuration ADDR[11:10].

Table 6.1.4 Relationship Among Different Clock Frequencies (for TMPR4938XBG-333, CPUCLK = 333 MHz)

Master Clock (Input) and Boot Configured Settings		Internal Clock			External Clock (Output)												
MASTERCLK (MHz)	Boot Configured Setting ADDR[3:0]	CPUCLK (MHz)	GBUSCLK (MHz)	IMBUSCLK (MHz)	SDCLK [3:0] (MHz)	SYSCLK (MHz) Boot Configured Settings PCIDIVMODE[2:0]				PCICLK[5:0] (MHz) † CCFG Settings PCIDIVMODE[2:0]							
						HH (1/1)	HL (1/2)	LH (1/3)	LL (1/4)	LLH (1/4)	LHH (1/4.5)	HLH (1/5)	HHH (1/5.5)	LLL (1/8)	LHL (1/9)	HLL (1/10)	HHL (1/11)
-	HHHH(x2.0)	333	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	HLHH(x8.0)																
133	HHHL (x2.5)		133	66	133	133	66	44	33	83	74	66	60	41	38	33	30
33	HLHL (x10.0)																
111	HHLH (x3.0)		111	55	111	111	55	37	27								
27	HLLH (x12.0)																
83	HHLL (x4.0)		83	41	83	83	41	27	20								
20	HLLL (x16.0)																
74	LHHH(x4.5)		74	37	74	74	37	24	18								
18	LLHH (x18.0)																

† The CCFG.PCIDIVMODE[2:1] field is setting by the boot configuration ADDR[11:10].

6.2 Power-Down Mode

6.2.1 Halt Mode and Doze Mode

The WAIT instruction causes the TX49/H3 core to enter either of the two low-power modes: Halt and Doze. The TX49/H3 can exit from Halt or Doze mode upon an interrupt exception. Ensure, therefore, that the TX49/H3 does not enter Halt or Doze mode when all interrupts are masked in the interrupt controller.

The HALT bit of the TX49/H3 core Config register is used to select Halt or Doze mode. As the TX4938 does not use the snoop function of the TX49/H3 core, the bit should be set to select Halt mode, which achieves greater power reduction than Doze mode.

6.2.2 Power Reduction for Peripheral Modules

When the system does not use the DMA controller, PCI controller, serial I/O controller, timers/counters, parallel I/O controller, or AC-link controller, it can stop the input clock for that module to reduce power dissipation.

The clock control register (CLKCTR) is used to control whether to turn each clock on or off. The module should be reset before its clock can be turned on or off. This reset is performed using the reset bit for the specific module, provided in the clock control register. The reset also initializes the registers of the module, thus requiring subsequent setup of necessary register values and other configurations. Refer to Section 5.2.5, "Clock Control Register" for detail of the clock control register (CLKCTR).

6.3 Power-On Sequence

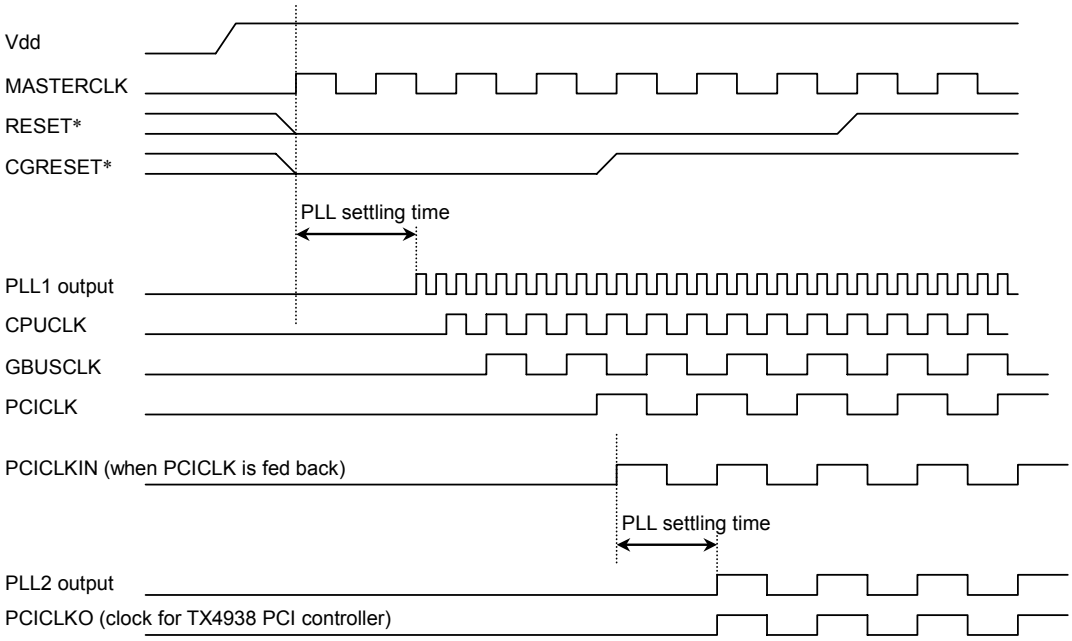


Figure 6.3.1 Power-On Sequence

7. External Bus Controller

7.1 Features

The External Bus Controller is used for accessing ROM, SRAM memory, and I/O peripherals. The features of this bus are described below.

(1) 8 independent channels

(2) Supports access to ROM (mask ROM, page mode ROM, EPROM, EEPROM), SRAM, flash memory, and I/O peripherals.

(3) Selectable data bus width of 8-bit, 16-bit, 32-bit for each channel

(4) Selectable full-speed, 1/2 speed, 1/3 speed, 1/4 speed for each channel

(5) Programmable timing for each channel. Programmable setup and hold time of address, chip enable, write enable, and output enable signals.

(6) Supports memory sizes from 1 MB to 1 GB for devices with a 32-bit data bus. Supports memory sizes from 1 MB to 512 MB for devices with a 16-bit data bus. Supports memory sizes from 1 MB to 256 MB for devices with an 8-bit data bus.

(7) Supports special DMAC Burst access (address decrement/fixed).

(8) Supports critical word first access of the TX49/H3 core.

(9) Supports page mode memory. Supports 4-, 8-, and 16-page size.

(10) Supports the External Acknowledge Signal (ACK*) and External Ready Signal modes.

(11) Channel 0 can be used as Boot memory. Boot settings can be made from the following selections:

- Data bus width: 8-bit, 16-bit, 32-bit
- ACK* output or ACK* input
- BWE pin (byte enable or byte Write enable)
- Boot channel clock frequency

(12) Supports the ISA I/O area access.

(13) Supports ATA PIO mode.

7.2 Block Diagram

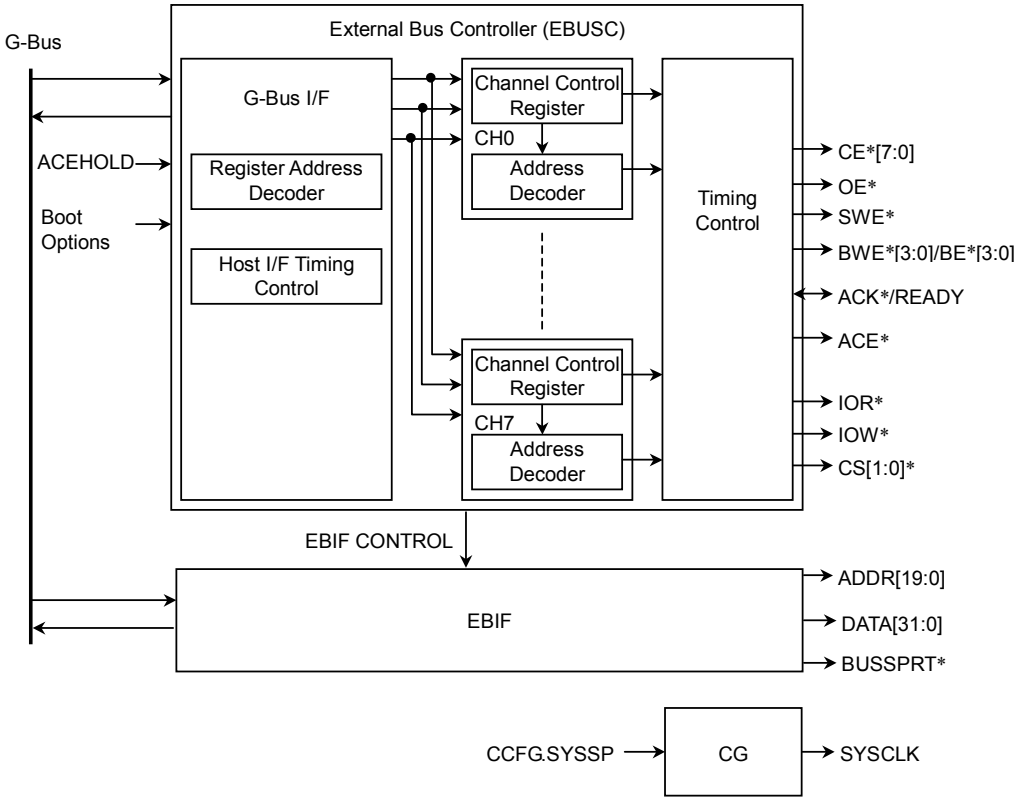


Figure 7.2.1 Block Diagram of External Bus Controller

7.3 Detailed Explanation

7.3.1 External Bus Control Register

The External Bus Controller (EBUSC) has eight channels. This register contains one Channel Control Register (EBCCRn) for each channel, and all settings can be made independently for each channel.

Either Word or Double-word access is possible for a Control Register. However, be sure to make any Enable settings to EBCCRn.ME last when using Word access and dividing register settings into two accesses. If EBCCRn.ME is enabled before setting the base address, then unintended memory access may result.

7.3.2 Global/Boot-up Options

In addition to the settings made separately for each channel, the Channel Control Registers can also use global options that make settings common to all channels.

External Bus Controller Channel 0 can be used as a Boot memory channel. Channel 0 is set by the external pins (Boot pins) during reset.

These settings are summarized below in Table 7.3.1. (Please refer to "3.2 Boot Configuration" and "5.2.1 Chip Configuration Register" for more information.)

Table 7.3.1 Global/Boot-up Options

Pin Name	Set Register	Explanation
—	CCFG.ARMODE	Selects the operation mode of the ACK*/READY signal. 0 = ACK*/READY Dynamic mode (Default) 1 = ACK*/READY Static mode
—	CCFG.ACEHOLD	Sets the address hold time relative to the ACE* signal. 0: Address changes simultaneous to deassertion of the ACE* signal. 1: Address changes 1 clock cycle after deassertion of the ACE* signal. (Default)
ADDR[14:13]	CCFG.SYSSP	Specifies the division ratio of the SYSCLK output relative to the internal bus clock (GBUSCLK). 00: 1/4 speed (1/4 the GBUSCLK frequency) 01: 1/3 speed (1/3 the GBUSCLK frequency) 10: 1/2 speed (1/2 the GBUSCLK frequency) 11: Full speed (same frequency as the GBUSCLK frequency)
ADDR[8]	EBCCR0.ME	Specifies whether to enable or disable Channel 0. 0: Disable this channel as a Boot channel. 1: Enable this channel as a Boot channel.
ADDR[7:6]	EBCCR0.SP	Specifies the operation speed of Channel 0. 00: 1/4 Speed mode 01: 1/3 Speed mode 10: 1/2 Speed mode 11: Full Speed mode
DATA[5]	EBCCR0.BC	When accessing Channel 0, specifies whether to use the BWE[3:0] signal as a Byte Enable signal (BE[3:0]) or to use it as a Byte Write Enable signal (BWE[3:0]). 0: Byte Enable mode 1: Byte Write Enable mode
DATA[4]	EBCCR0.WT[0]	Specifies the Channel 0 access mode. 0: Normal mode (DATA[4] = H) 1: External ACK mode (DATA[4] = L)
DATA[1:0]	EBCCR0.BSZ	Specifies the memory bus width of Channel 0. 00: Reserved 01: 32-bit width 10: 16-bit width 11: 8-bit width

7.3.3 Address Mapping

Each of the eight channels can use the Base Address field (EBCCRn.BA[35:20]) and the Channel Size field (EBCCRn.CS[3:0]) of the External Bus Channel Control Register to map to any physical address.

A channel is selected when the following equation becomes True.

paddr[35:20] & !Mask[35:20] == BA[35:20] & !Mask[35:20]

In the above equation, paddr represents the accessed physical address, Mask[35:20] represents the address mask value selected from Table 7.3.2 from the Channel Size field value, the ampersand (&) represents the AND operation, and the exclamation mark (!) represents the Logical NOT for each bit.

Operation is indeterminate when either multiple channels are selected simultaneously, or a channel is selected simultaneously with the SDRAM Controller or PCI Controller.

Table 7.3.2 Address Mask

CS[3:0]	Channel Size	Address Mask[35:20]
0000	1 MB	0000_0000_0000_0000
0001	2 MB	0000_0000_0000_0001
0010	4 MB	0000_0000_0000_0011
0011	8 MB	0000_0000_0000_0111
0100	16 MB	0000_0000_0000_1111
0101	32 MB	0000_0000_0001_1111
0110	64 MB	0000_0000_0011_1111
0111	128 MB	0000_0000_0111_1111
1000	256 MB	0000_0000_1111_1111
1001	512 MB	0000_0001_1111_1111
1010	1 GB	0000_0011_1111_1111
1011	Reserved	Reserved
1100	Reserved	Reserved
1101	Reserved	Reserved
1110	Reserved	Reserved
1111	Reserved	Reserved

7.3.4 External Address Output

The maximum memory space size for each channel is 1 GB (230B). Addresses are output by dividing the 20-bit ADDR[19:0] signal into two parts: the upper address and the lower address. The address bit output to each bit of the ADDR[19:0] signal changes according to the setting of the channel data bus width. (See "7.3.5 Data Bus Size" for more information.)

It is possible for an external device to latch the upper eight address bits using the ACE* signal. Either the ACE* signal itself can be used as a Latch Enable signal or the upper address can be latched at the rise of SYSCLK when the ACE* signal is being asserted.

The ADDR signal output is held for one clock cycle after the ACE* signal rise when the CCFG.ACEHOLD bit is set (default). (See Figure 7.5.1.) The ADDR signal output is not held when the CCFG.ACEHOLD bit is cleared. This hold time setting is applied globally to all channels.

The ACE* signal of the upper address is always asserted at the first external bus access cycle after Reset. In all subsequent external bus access cycles, the bit mapping of the upper address output to ADDR[19:12] is compared to the bit mapping of the upper address output to ADDR[19:12] previously. The upper address is output and the ACE* signal is asserted only if the compared results do not match.

As indicated below in Table 7.3.3, in the case of channel sizes that do not use the upper address latched by the ACE* signal, with the exception of the first cycle after reset, the upper address is not output and the ACE* signal is not asserted.

Table 7.3.3 Relationship Between the Upper Address Output and the Channel Size (CS)

CS / Bus Width	1 MB	2 MB	4 MB	8 MB or more
32 bits	—	—	—	√
16 bits	—	—	√	√
8 bits	—	√	√	√

√: The upper address output changes when the upper address changes.

—: The upper address output does not change (with the exception of the first cycle after reset.)

7.3.5 Data Bus Size

The External Bus Controller supports devices with a data bus width of 8 bits, 16 bits, and 32 bits. The data bus width is selected using the BSZ field of the Channel Control Register (EBCCRn). The address bits output to each bit of the ADDR[19:0] signal change according to the mode. When access of a size larger than the data bus width is performed, the dynamic bus sizing function is used to execute multiple bus access cycles in order from the lower address.

7.3.5.1 32-bit Bus Width Mode

DATA[31:0] becomes valid.

Bits [21:2] of the physical address are output to ADDR[19:0]. The internal address bits [29:22], which are the upper address, are multiplexed to external ADDR[19:12]. The maximum memory size is 1 GB.

Table 7.3.4 Address Output Bit Correspondence in the 32-bit Mode

ADDR Bit	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper Address	29	28	27	26	25	24	23	22												
Lower Address	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

When a Single cycle that accesses 1-Byte/1 half-word/1-word data is executed, 32-bit access is executed only once on the external bus. 32-bit access is executed twice when performing 1-double-word access. When a Burst cycle is executed, two 32-bit cycles are executed for each Burst access when the Bus cycle tries to request a byte combination other than double-word data.

7.3.5.2 16-bit Bus Width Mode

DATA[15:0] becomes valid.

Bits [20:1] of the physical address are output to ADDR[19:0]. The internal address bits [28:21], which are the upper address, are multiplexed to external ADDR[19:12]. In other words, the address is shifted up one bit relative to the 32-bit bus mode when output. As a result, the maximum memory size of the 16-bit bus mode is 512 MB.

Table 7.3.5 Address Output Bit Correspondence in the 16-bit Mode

ADDR Bit	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper Address	28	27	26	25	24	23	22	21												
Lower Address	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

When a Single cycle that accesses 1-Byte or 1 half-word data is executed, 16-bit access is executed only once on the external bus. 16-bit access is executed twice when performing 1-word access. 16-bit access is executed four times when performing 1-double-word access. When a Burst cycle is executed, four 16-bit cycles are executed for each Burst access when the Bus cycle tries to request a byte combination other than double-word data.

7.3.5.3 8-bit Bus Width Mode

DATA[7:0] becomes valid.

Bits [19:0] of the physical address are output to ADDR[19:0]. The internal address bits [27:20], which are the upper address, are multiplexed to external ADDR[19:12]. In other words, the address is shifted up two bits or more relative to the 32-bit bus mode when output. As a result, the maximum memory size of the 8-bit bus mode is 256 MB.

Table 7.3.6 Address Output Bit Correspondence in the 8-bit Mode

ADDR Bit	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper Address	27	26	25	24	23	22	21	20												
Lower Address	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When a Single cycle that accesses 1-Byte data is executed, 8-bit access is executed only once on the external bus. 8-bit access is executed twice when performing 1-half-word access. 8-bit access is executed four times when performing 1-word access. 8-bit access is executed eight times when performing 1-double-word access. When a Burst cycle is executed, eight 8-bit cycles are executed for each Burst access when the Bus cycle tries to request a byte combination other than double-word data.

7.3.6 Access Mode

The following four modes are available as controller access modes. These modes can be set separately for each channel.

- Normal mode
- Page mode
- External ACK mode
- Ready mode

Depending on the combination of modes in each channel, either of two modes in which the ACK*/Ready signal operates differently (ACK*/Ready Dynamic mode, ACK*/Ready Static mode) is selected by the ACK*/Ready Mode bit (CCFG.ARMODE) of the Chip Configuration Register. The mode selected is applied globally to all channels.

(1) ACK*/READY Dynamic mode (CCFG.ARMODE = 0)
This mode is selected in the initial state.

The ACK*/Ready signal automatically switches to either input or output according to the setting of each channel. When in the Normal mode or the Page mode, the ACK*/Ready signal is an output signal, and the internally generated ACK* signal is output. When in the External ACK* or Ready mode, the ACK*/Ready signal becomes an input signal. The ACK*/Ready signal outputs High if there is no access to the External Bus Controller. However, this signal may output Low during access to SDRAM.

Please refer to the timing diagrams (Figure 7.5.23 and Figure 7.5.24) and be careful to avoid conflicts when switching from output to input.

(2) ACK*/Ready Static mode (CCFG.ARMODE = 0)

The internally generated ACK* signal is not output when in either the Normal mode or Page mode. Therefore, the ACK*/Ready signal will not become an output in any channel.

Access using Burst transfer by the internal bus (G-Bus) is supported when in a mode other than the Ready mode. However, the Ready mode is not supported.

Table 7.3.7 Operation Mode

	PM	RDY	PWT:WT	Mode	ACK*/READY Pin State	Access End Timing State	G-Bus Burst Access
ACK*/Ready Dynamic Mode	0	0	!3f	Normal	Output	Internally Generated ACK*	√
			3f	External ACK*	Input	ACK* Input	√
		1	—	READY	Input	Ready Input	—
	!0	0	—	Page	Output	Internally Generated ACK*	√
		1	—	Reserved	—	—	—
ACK*/Ready Static Mode	0	0	!3f	Normal	Hi-Z	Internally Generated ACK*	√
			3f	External ACK*	Input	ACK* Input	√
		1	—	READY	Input	Ready Input	—
	!0	0	—	Page	Hi-Z	Internally Generated ACK*	√
		1	—	Reserved	—	—	—

7.3.6.1 Normal Mode

When in this mode, the ACK*/Ready signal becomes an ACK* output when it is in the ACK*/Ready Dynamic mode. The ACK*/Ready signal becomes High-Z when it is in the ACK*/Ready Static mode.

Wait cycles are inserted according to the EBCCRn.PWT and EBCCRn.WT value at the access cycle. The Wait cycle count is 0 – 0x3e (becomes the external ACK mode when set to EBCCRn.PWT: WT = 0x3f).

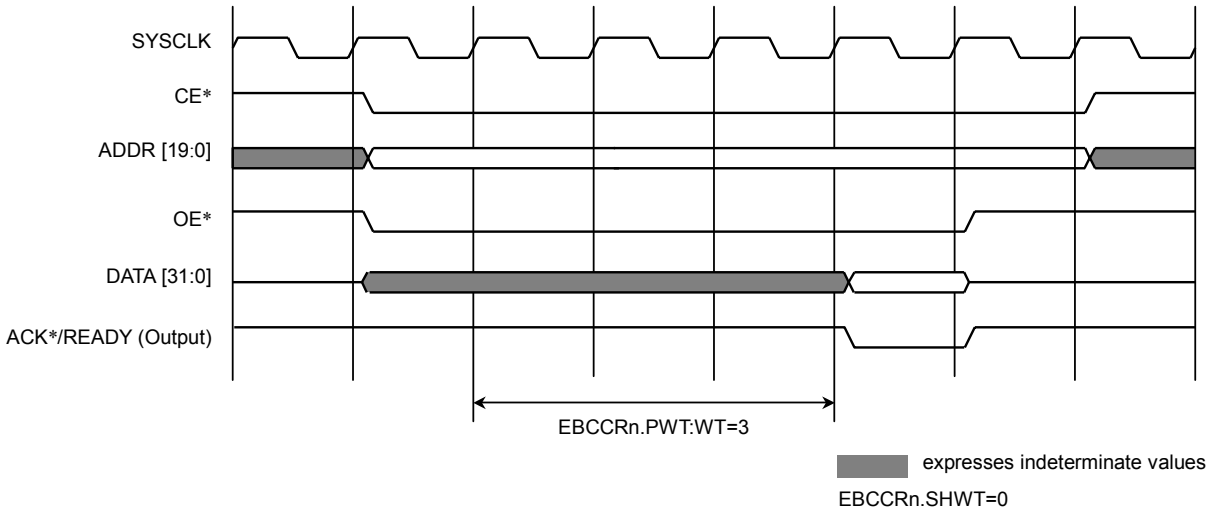


Figure 7.3.1 Normal Mode

7.3.6.2 External ACK Mode

When in this mode, the ACK*/READY pin becomes ACK* input, and the cycle is ended by the ACK* signal from an external device. ACK* input is internally synchronized. Refer to Section "7.3.7.4 ACK* Input Timing" for more information regarding timing.

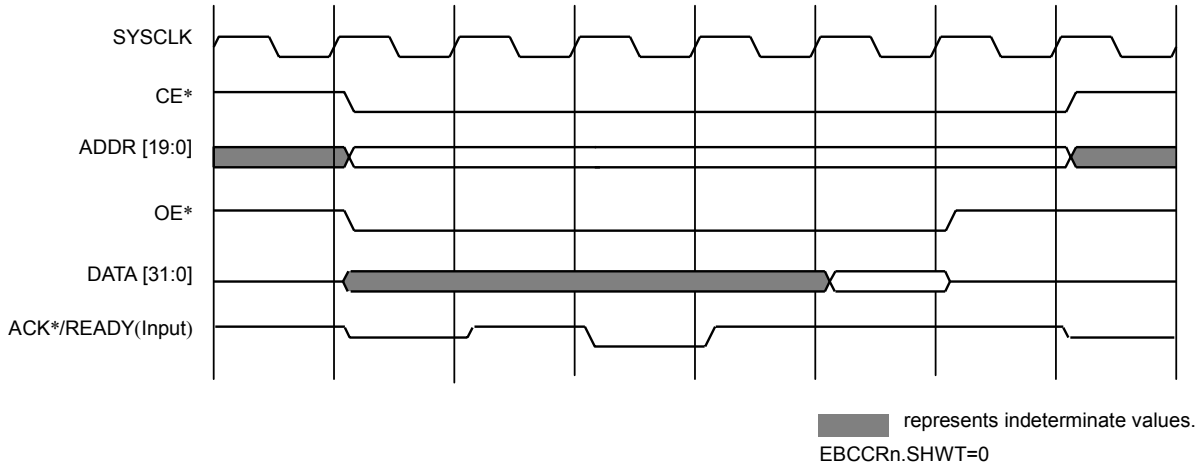


Figure 7.3.2 External ACK Mode

7.3.6.3 Ready Mode

When in this mode, the ACK*/Ready pin becomes Ready input, and the cycle is ended by Ready input from an external device. Ready input is internally synchronized. See Section "7.3.7.5 Ready Input Timing" for more information regarding the operation timing.

When the Wait cycle count specified by EBCCREBCCRn.PWT:WT elapses, a check is performed to see whether the Ready signal was asserted. Since EBCCRn.WT[0] is used to indicate the ACK*/ Ready Static/Dynamic mode, it is not used for setting the Wait cycle count. Therefore, the Wait cycle count that can be set by the Ready mode is 0, 2, 4, 6, …, 62.

When the number of wait cycles is 0, READY check is started in 1 cycle after asserting the CE* signal. When the number of wait cycles is other than zero, after waiting only for the specified number of cycles, READY check is started.

The Ready mode does not support Burst access by the internal bus.

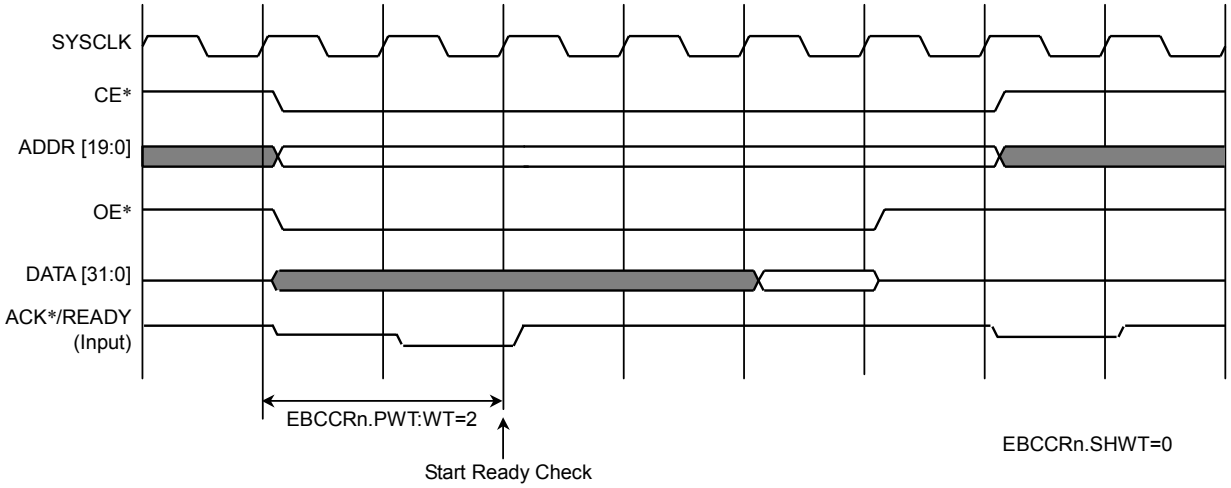


Figure 7.3.3 Ready Mode

7.3.6.4 Page Mode

When in this mode, the ACK*/Ready pin becomes ACK* output when it is in the Dynamic mode. When it is in the ACK*/Ready Static mode, the ACK*/Ready signal becomes HiZ.

Wait cycles are inserted into the access cycle according to the values of EBCCRn.PWT and EBCCRn.WT. The Single access protocol in Page mode is identical to that of Normal mode, except the number of wait cycles inserted. The Wait cycle count in the first access cycle of Single access or Burst access is determined by the EBCCRn.WT value. The Wait cycle count can be set from 0 to 15. The Wait cycle count of subsequent Burst cycles is determined by the EBCCRn.PWT value. The Wait cycle count can be set from 0 to 3.

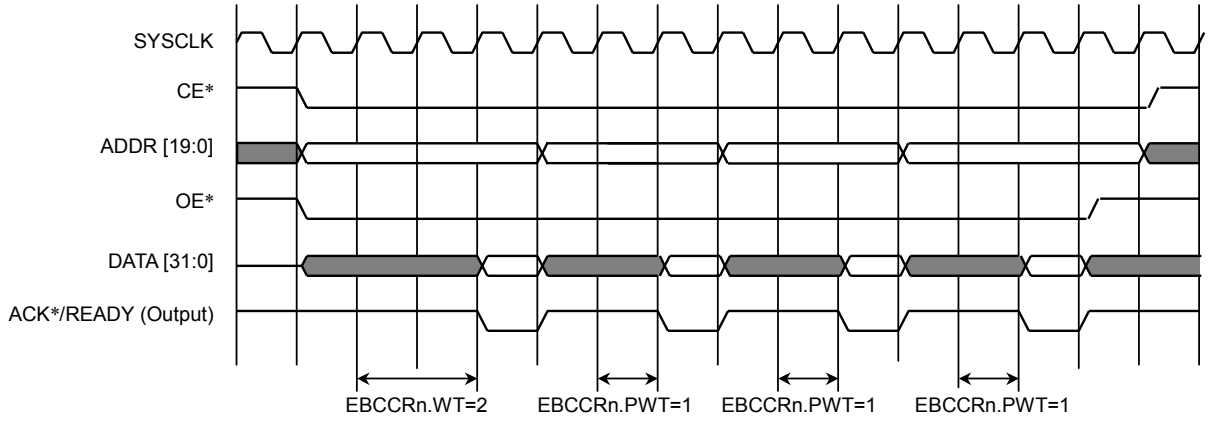


Figure 7.3.4 Page Mode (Burst Access)

7.3.7 Access Timing

7.3.7.1 SHWT Option

The SHWT option is selected when the SHWT (Setup/Hold Wait Time) field of the Channel Control Register is a value other than "0." This option inserts Setup cycles and Hold cycles between signals as follows.

Setup cycle: CE* from ADDR, OE* from CE*, BWE* from CE*, SWE* from CE*.

Hold cycle: ADDR from CE*, CE* from OE*, CE* from BWE*, CE* from SWE*

This option is used for I/O devices that are generally slow. All Setup cycles and Hold cycles will be identical, so each cycle cannot be set individually.

The SHWT mode cannot be used by the Page mode. The SHWT mode can be used by all other modes, but there is one restriction: the internal bus cannot use Burst access.

The hold cycles of DATA relative to SWE* and BWE* are fixed at one clock cycle, regardless of the settings of the SHWT option. When the SHWT option is disabled, the setup cycles of SEW* and BWE* relative to CE*, and the hold cycles of OE*, SEW* and BWE* relative to CE* are one clock cycle.

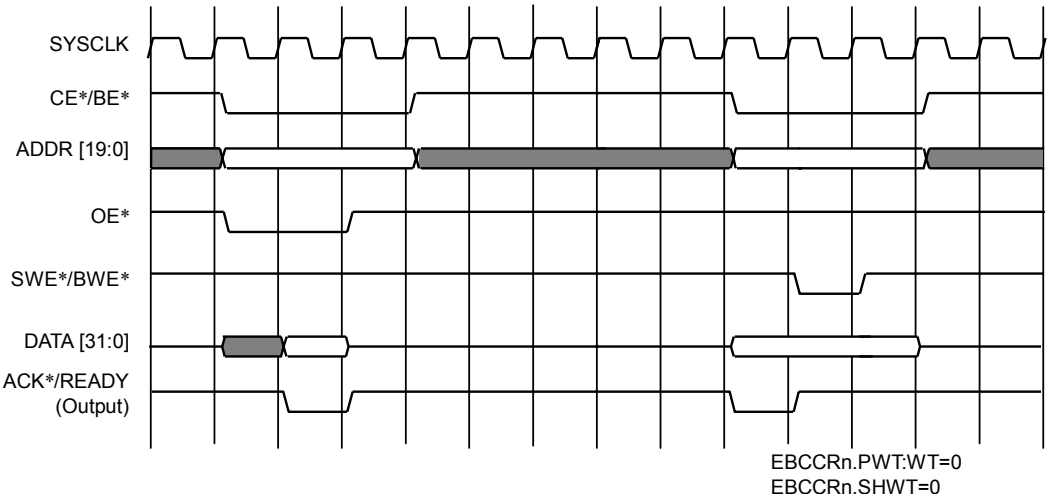


Figure 7.3.5 SWHT Disable (Normal Mode, Single Read/Write Cycle)

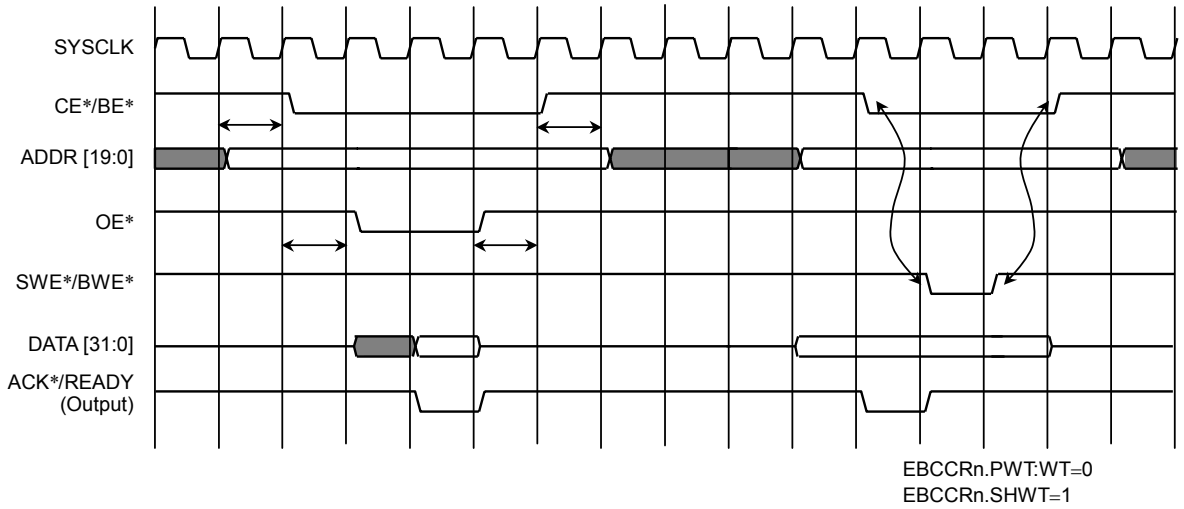


Figure 7.3.6 SHWT 1 Wait (Normal Mode, Single Read/Write Cycle)

7.3.7.2 ACK*/READY Input/Output Switching Timing

When in the ACK*/Ready Static mode, the ACK*/Ready signal is always an input signal. When in the ACK*/Ready Dynamic mode, the ACK*/Ready signal is an input signal when in the External ACK mode or the Ready mode, but is an output signal in all other modes.

During External ACK mode or Ready mode access, the ACK* signal becomes High-Z at the cycle where the CE* signal is asserted. At the end of the access cycle, the ACK* signal is output (driven) again one clock cycle after the CE* signal is deasserted (see Figure 7.3.3 and Figure 7.5.23).

7.3.7.3 ACK* Output Timing (Normal Mode, Page Mode)

When in the Normal mode and Page mode of the ACK*/Ready Dynamic mode, the ACK* signal becomes an output signal and is asserted for one clock cycle to send notification to the external device of the data Read and data Write timing.

During the Read cycle, the data is latched at the rise of the next clock cycle after when the ACK* signal is asserted. (See Figure 7.3.7 ACK* Output Timing (Single Read Cycle)).

During the Write cycle, SWE*/BWE* is deasserted at the next clock cycle after when the ACK* signal is deasserted, and the data is held for one more clock cycle after that. (See Figure 7.3.8 ACK* Output Timing (Single Write Cycle)).

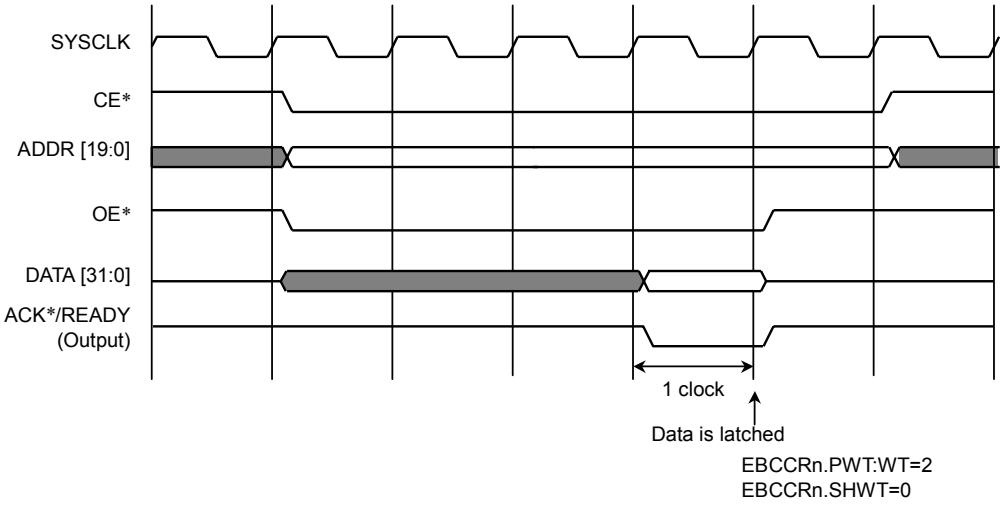


Figure 7.3.7 ACK* Output Timing (Single Read Cycle)

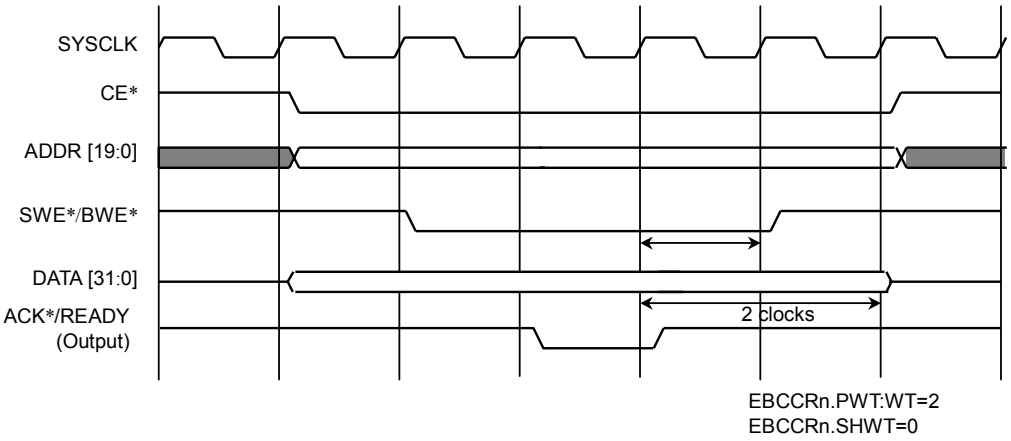


Figure 7.3.8 ACK* Output Timing (Single Write Cycle)

7.3.7.4 ACK* Input Timing (External ACK Mode)

The ACK* signal becomes an input signal when in the external ACK mode.

During a Read cycle, data is latched two clock cycles after assertion of the ACK* signal is acknowledged (Figure 7.3.9 ACK* Input Timing (Single Read Cycle)). During a Write cycle, assertion of the ACK* signal is acknowledged, SWE*/BWE* is deasserted three clock cycles later, then data is held for one clock cycle after that (Figure 7.3.10 ACK* Input Timing (Single Write Cycle).

The ACK* input signal is internally synchronized. Due to internal State Machine restrictions, ACK* cannot be acknowledged consecutively on consecutive clock cycles. External devices can assert ACK* across multiple clock cycles under the following conditions.

- During Single access, the ACK* signal can be asserted before the end of the cycle during which CE* is dasserted.
- During Burst access, it is possible to assert the ACK* signal for up to three clock cycles during Reads and for up to five clock cycles during Writes. If the ACK* signal is asserted for a period longer than this, it will be acknowledged as the next valid ACK* signal.

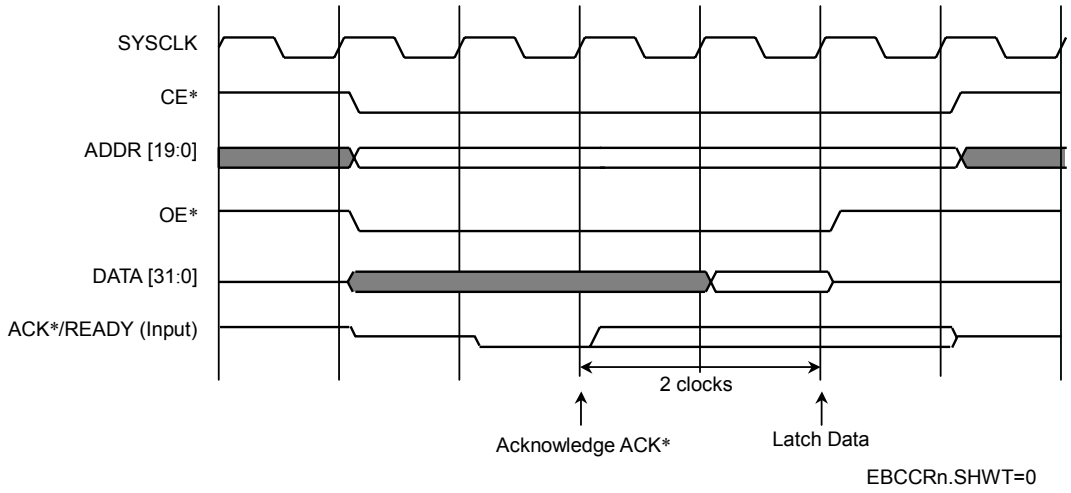


Figure 7.3.9 ACK* Input Timing (Single Read Cycle)

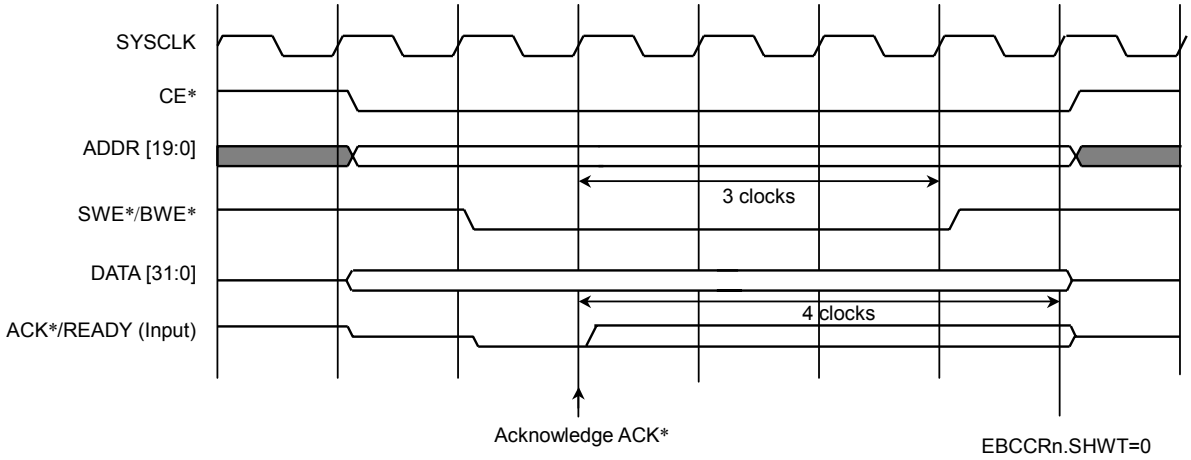


Figure 7.3.10 ACK* Input Timing (Single Write Cycle)

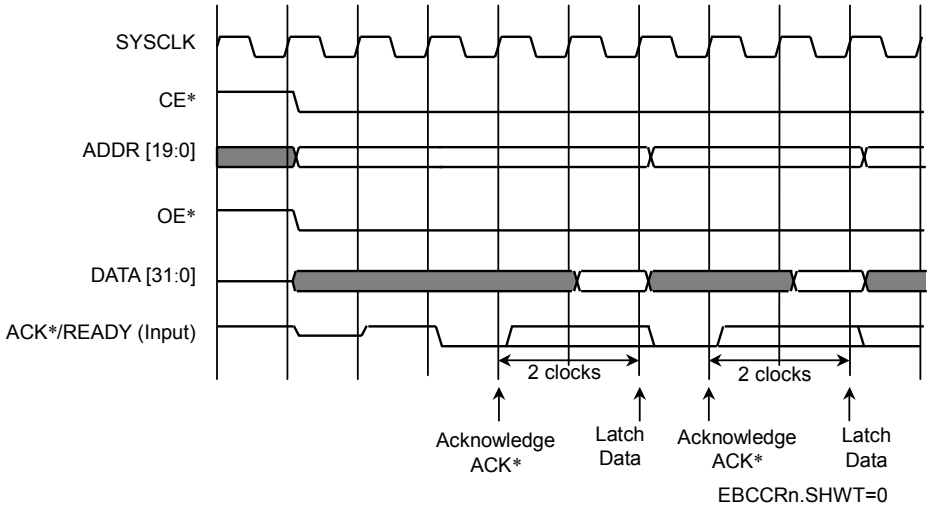


Figure 7.3.11 ACK* Input Timing (Burst Read Cycle)

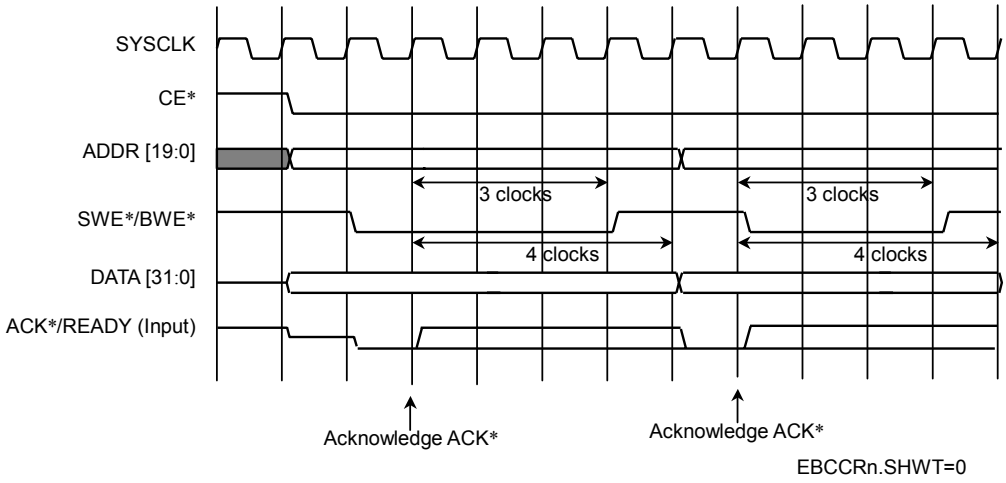


Figure 7.3.12 ACK* Input Timing (Burst Write Cycle)

7.3.7.5 Ready Input Timing

The ACK*/Ready pin is used as a Ready input when in the Ready mode. The Ready input timing is the same as the ACK* input timing explained in 7.3.7.4 ACK* Input Timing (External ACK Mode) with the two following exceptions.

- Ready must be a High Active signal.
- When in the Ready mode, the Wait cycle count specified by EBCCRn.PWT:WT must be inserted in order to delay the Ready signal check (see 7.3.6.3 Ready Mode).

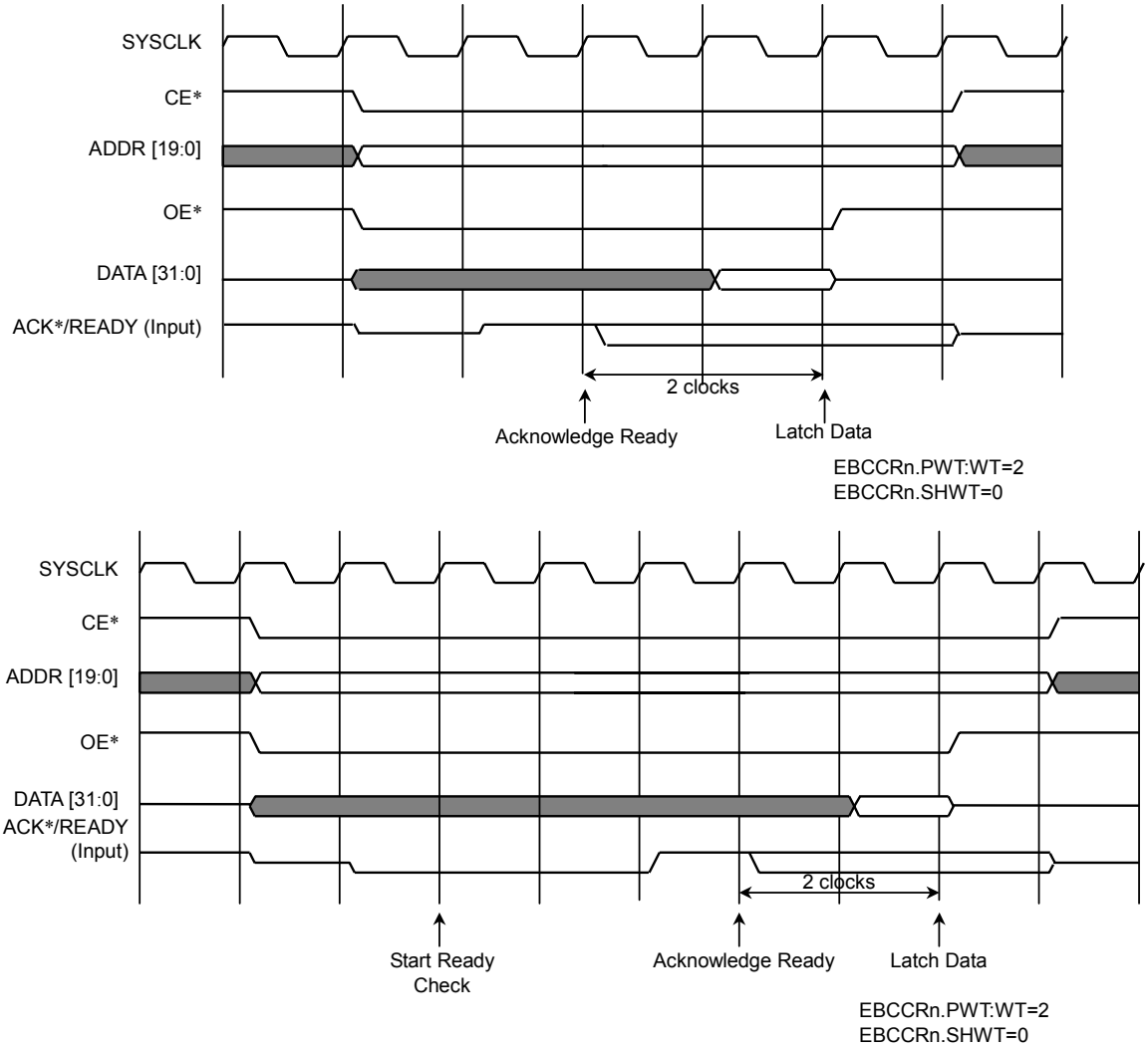


Figure 7.3.13 Ready Input Timing (Read Cycle)

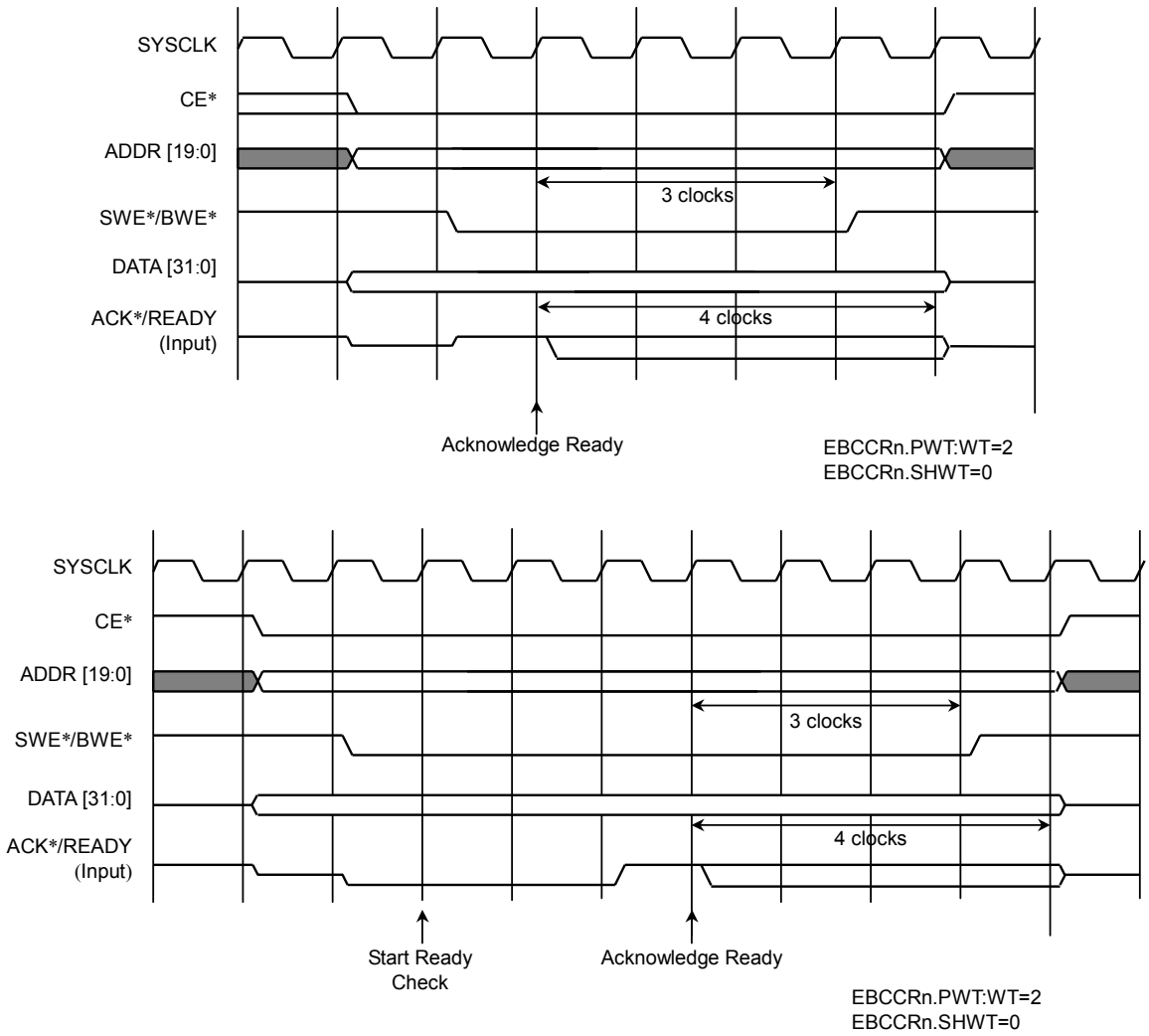


Figure 7.3.14 Ready Input Timing (Write Cycle)

7.3.8 Clock Options

External devices connected to the external bus can use the SYSCLK signal as the clock. The SYSCLK signal clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK): 1/1, 1/2, 1/3, 1/4. The ADDR[14:13] signal is used to set this frequency during reset, and the setting is reflected in the SYSCLK Division Ratio field (CCFG.SYSSP) of the Chip Configuration Register.

The operation reference clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK) for each channel independent of the SYSCLK signal clock frequency: 1/1, 1/2, 1/3, 1/4. The external signal of the External Bus Controller operates synchronous to this operation clock. The Bus Speed field (EBCCRn.SP) of the External Bus Channel Control Register sets this frequency.

Please set the same value as CCFG.SYSSP to EBCCRn.SP when the external device uses the SYSCLK signal. If these two values do not match, then the channel, the operation reference clock, and the SYSCLK signal will no longer be synchronous and will not operate properly.

7.3.9 ISA /ATA Mode

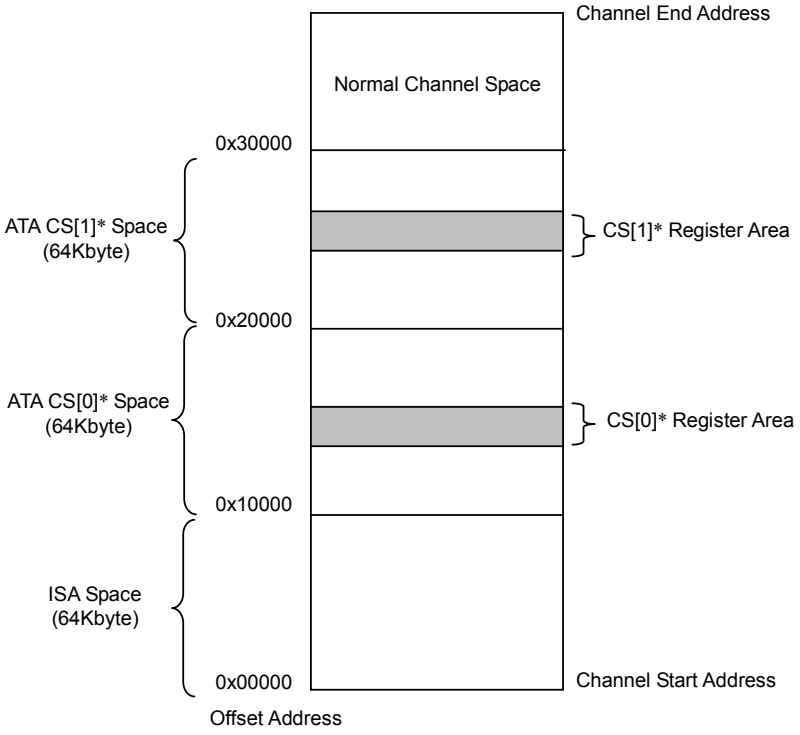
TMPR4938 supports ISA I/O space access and ATA PIO transfer mode. Since the pins used in ISA/ATA mode are multiplexed pins, select ISA/ATA before use of these pins. Since the signal of CE* is not used in this mode, when ETHER1 is chosen, this mode can be used by channels 5, 6 and 7. (Refer to “5.2.3 Pin Configuration Register”.)

7.3.9.1 Address space

In the channel set to EBCCRn.ISA=01, 64 kbyte from the start address of the channel is used for the ISA I/O space. (Refer to Figure 7.3.15). When access to the ISA I/O space is performed, IOR* and IOW* control signals are valid, and OE*, SWE* and CE* are invalid. Other control signals are usable, that is the same as the case other than the ISA/ATA mode.

In the channel set to EBCCRn.ISA=11, 64 kbyte from the start address of the channel is used for ISA I/O space, and successive 128 kbyte is used for ATA space (64 kbyte each for CS[0]* and CS[1]*). (Refer to Figure 7.3.15). When access to the ATA space is performed, the IOR*, IOW* and CS* control signals are valid, and OE*, SWE* and CE* are invalid. Other control signals are usable, that is the same as the case other than the ISA/ATA mode.

In general, the space from address off set 0x30000 (0x10000 in the ISA mode) to the channel end address becomes the channel space. Access to this space is the same as the case other than the ISA/ATA mode. (OE*, SWE* and CE* control signals are valid, and IOR*, IOW* and CS* are invalid).



When EBCCRn.ISA is set to 01, the ATA CS[0]* and CS[1]* spaces become the normal channel space.

See Table 7.3.8 for the mapping of the ATA CS[1]* and CS[0]* register areas.

Figure 7.3.15 Physical Address Mapping of the ISA/ATA Spaces

7.3.9.2 Usage Considerations

To use ISA and ATA modules in READY, their ICCHRDY or IORDY pin should be connected to the ACK*/READY pin of the TX4938.

The SHWT option allows users to adjust the setup and hold timing of IOR* and IOW* relative to ADDR.

When the READY mode and SHWT option are used, the internal bus burst accesses are prohibited.

Although the ISA and ATA signals are 5-volt signals, the TX4938 is not 5-volt-tolerant. For voltage-level conversion, bus switches such as BUSSW and LCX245 must be inserted between ISA busses according to the modules used.

7.3.9.3 Usage Examples

(1) 8-bit ISA module

Following is a programming example to use an 8-bit ISA module:

- ISA mode (EBCCRn.ISA=01)
- 8-bit bus (EBCCRn.BSZ=11)
- READY mode (EBCCRn.RDY=1, EBCCRn.PM=00)
- Channel enable (EBCCRn.ME=1)
- ISA select (PCFG.ISA_SEL=1)

Program the number of wait cycles, the SHWT option and the bus speed according to the ISA module used. Figure 7.3.16 shows an example of connecting an 8-bit ISA module to the TX4938.

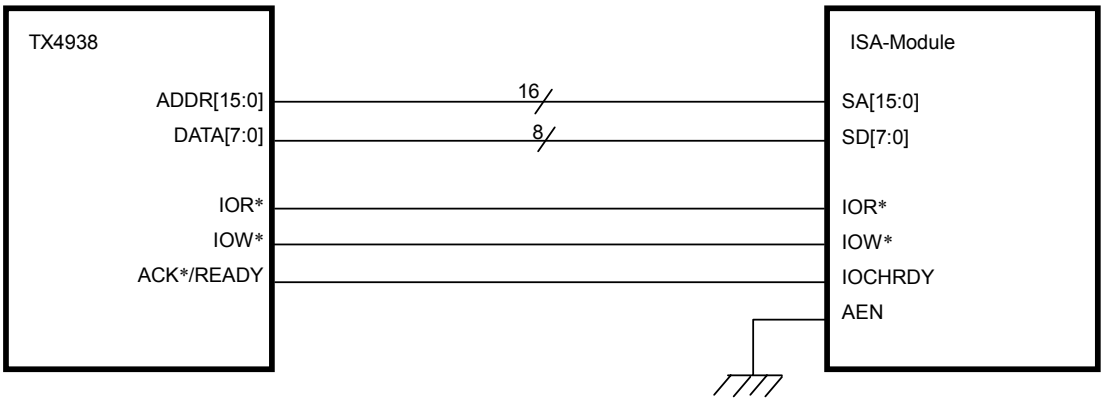


Figure 7.3.16 An Example of Connecting an 8-Bit ISA Module

(2) 16-bit ISA module

Following is a programming example to use a 16-bit ISA module:

- ISA mode (EBCCRn.ISA=01)
- 16-bit bus (EBCCRn.BSZ=10)
- READY mode (EBCCRn.RDY=1, EBCCRn.PM=00)
- Channel enable (EBCCRn.ME=1)
- ISA select (PCFG.ISA_SEL=1)

Program the number of wait cycles, the SHWT option and the bus speed according to the ISA module used. Figure 7.3.17 shows an example of connecting a 16-bit ISA module to the TX4938.

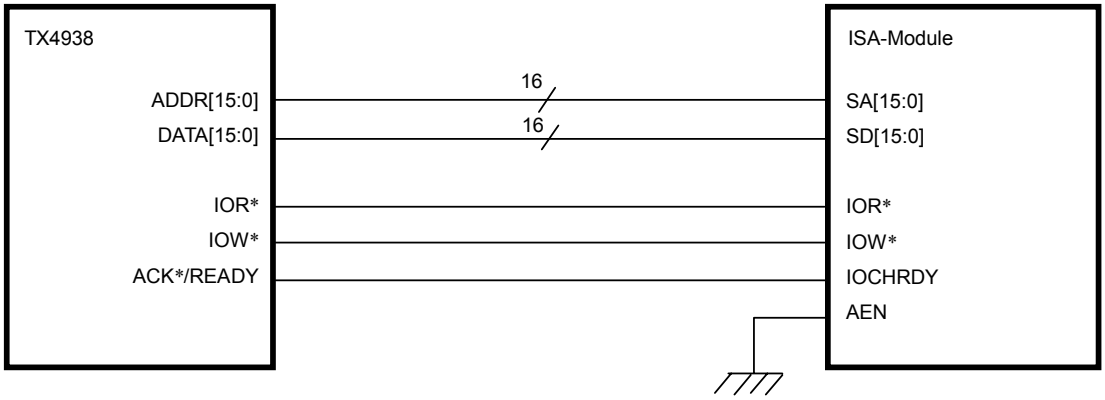


Figure 7.3.17 An Example of Connecting an 16-Bit ISA Module

With the EBUSC configured for a 16-bit bus, bits [16:1] of physical addresses are driven out from ADDR[15:0]. Thus, the address pins must be connected, as shown in Figure 7.3.17. To access halfwords in the ISA module with even addresses (*0, *2, *4, *6, …), the TX4938 must provide physical addresses with two low-order bits cleared (*0, *4, *8, *C, …). Therefore, the accessible ISA address space is limited to a maximum of 32 kbytes between 0x0000 and 0x7fff.

For byte accesses, the valid byte lane in the data bus differs, depending on whether the TX4938 is configured for big-endian or little-endian byte ordering. Software is responsible for address conversion for a specific endianness, as shown in Table 7.3.8.

(3) ATA

Following is a programming example to use an ATA module:

- ISA/ATA mode (EBCCRn=11)
- 16-bit bus (EBCCRn=10)
- READY mode (EBCCRnRDY=1, EBCCRN.PM=00)
- Channel enable (EBCCRn.ME=1)
- ISA/ATA pin select mode (PCFG.ISA_SEL=1, PCFG.ATA_SEL=1)

Program the number of wait cycles, the SHWT option and the bus speed according to the ISA module used. Figure 7.3.18 shows an example of connecting a 16-bit ISA module to the TX4938.

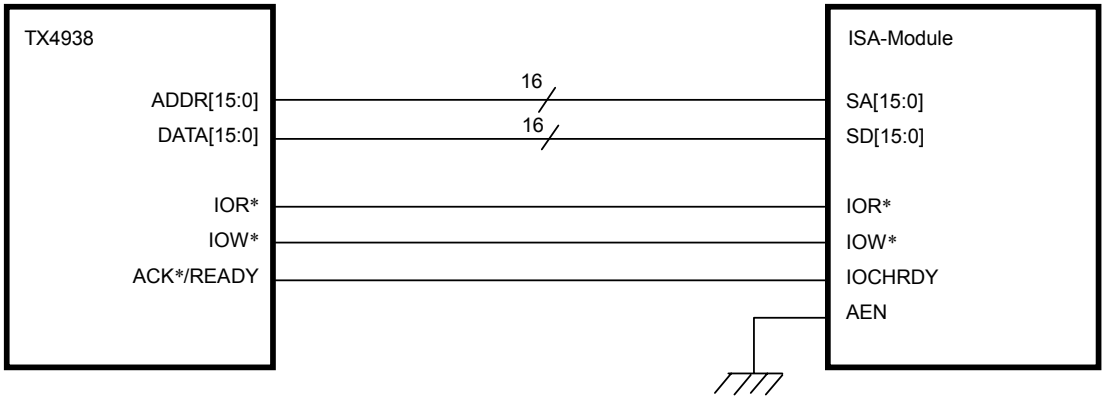


Figure 7.3.18 An Example of Connecting an ATA Module

Since the ATA has a 16-bit data bus, the same considerations apply to the ATA module as for the 16-bit ISA module. With the EBUSC configured for a 16-bit bus, bits [3:1] of physical addresses are driven out from ADDR[2:0]. Table 7.3.8 shows the relationships in offset addresses between the ATA and the TX4938.

For byte accesses, the valid byte lane in the data bus differs, depending on whether the TX4938 is configured for big-endian or little-endian byte ordering. To access 8-bit registers in the ATA address space, address conversion is required, as shown in Table 7.3.8.

Table 7.3.8 Relationships Between the ATA Addresses and Physical Addresses

	ATA Address Offset	ATA Register Mnemonic	TX4938 Physical Address Offset		
			16-Bit Access	8-Bit Access	
				Little-Endian	Big-Endian
CS[0]* Space	0x1F0	DTR (16bit)	0x103E0	-	-
	0x1F1	ERR/FTR (8bit)	-	0x103E2	0x103E3
	0x1F2	SCR (8bit)	-	0x103E4	0x103E5
	0x1F3	SNR (8bit)	-	0x103E6	0x103E7
	0x1F4	CLR (8bit)	-	0x103E8	0x103E9
	0x1F5	CHR (8bit)	-	0x103EA	0x103EB
	0x1F6	DHR (8bit)	-	0x103EC	0x103ED
	0x1F7	STR/CMR (8bit)	-	0x103EE	0x103EF
CS[1]* Space	0x3F6	ASR/DCR (8bit)	-	0x207EC	0x207ED

7.4 Register

Table 7.4.1 External Bus Controller (EBUSC) Registers

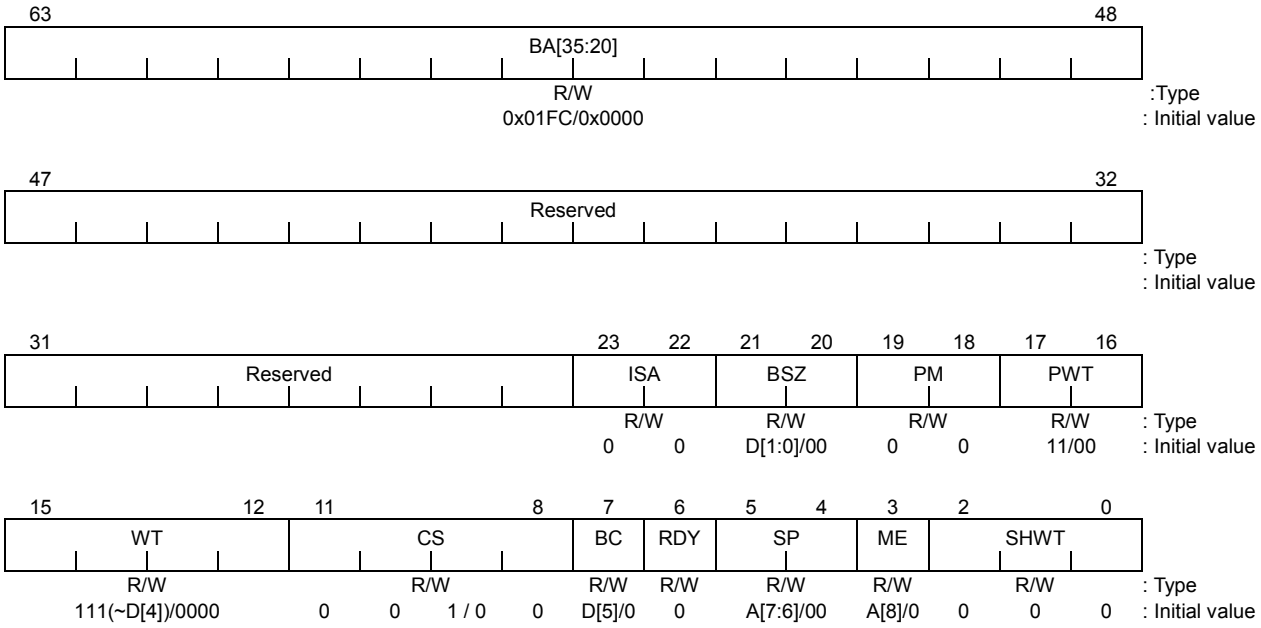
Offset Address	Bit Width	Register Symbol	Register Name
0x9000	64	EBCCR0	E-Bus Channel Control Register 0
0x9008	64	EBCCR1	E-Bus Channel Control Register 1
0x9010	64	EBCCR2	E-Bus Channel Control Register 2
0x9018	64	EBCCR3	E-Bus Channel Control Register 3
0x9020	64	EBCCR4	E-Bus Channel Control Register 4
0x9028	64	EBCCR5	E-Bus Channel Control Register 5
0x9030	64	EBCCR6	E-Bus Channel Control Register 6
0x9038	64	EBCCR7	E-Bus Channel Control Register 7

7.4.1 External Bus Channel Control Register (EBCCRn)

0x9000 (ch. 0), 0x9008 (ch. 1)
0x9010 (ch. 2), 0x9018 (ch. 3)
0x9020 (ch. 4), 0x9028 (ch. 5)
0x9030 (ch. 6), 0x9038 (ch. 7)

Channel 0 can be used as Boot memory. Therefore, the default is set by the Boot signal (see 7.3.2 Global/Boot-up Options). Channels 1 - 7 have the same register configuration as Channel 0, but they have different defaults than Channel 0.

When the EBCCRn is programmed using a sequence of 32-bit store instructions, the base address in the high-order 32-bit portion of the register must be written first, followed by the Master Enable bit in the low-order 32-bit portion.



Only in the case of Channel 0 are fields with different defaults in the "Channel 0/Other channel" state. D[ ] represents the corresponding Data[ ] signal value when the RESET* signal is deasserted. A[ ] represents the corresponding ADDR[ ] signal value when the RESET* signal is deasserted.

Bit	Mnemonic	Field Name	Description	Read/Write
63:48	BA[35:20]	Base Address	External Bus Control Base Address (Default: 0x01FC/0x0000) A physical address is used to specify the base address. The upper 16 bits [35:20] of the physical address are compared to the value of this field.	R/W
47:24		Reserved		—
23:22	ISA	ISA/ATA Mode	ISA/ATA Mode (Default: 00) Specifies the ISA I/O mode and ATA PIO mode. 00: ISA/ATA Mode disable 10: Reserved 01: ISA Mode enable 11: ATA Mode enable	R/W
21:20	BSZ	Bus Width	External Bus Control Bus Size (Default: DATA[1:0]/00) Specifies the memory bus width. 00: Reserved 10: 16-bit width 01: 32-bit width 11: 8-bit width Note: DATA[1:0] is set to Channel 0 as the default.	R/W

Figure 7.4.1 External Bus Channel Control Register (1/3)

Bit	Mnemonic	Field Name	Description	Read/Write
19:18	PM	Page Mode Page Size	External Bus Control Page Mode Page Size (Default: 00) Specifies the Page mode (Page mode memory support) use and page size. 00: Normal mode 01: 4-page mode 10: 8-page mode 11: 16-page mode	R/W
17:16	PWT	Page Mode Wait time	External Bus Control Page Mode Wait Time (Default: 11 / 00) Specifies the wait cycle count during Burst access when in the Page mode. 00: 0 wait cycles 10: 2 wait cycles 01: 1 wait cycle 11: 3 wait cycles Specifies a wait cycle count from 0 to 62 that matches WT when in the Normal mode or Ready mode. (See the WT item.)	R/W
15:12	WT	Normal Mode Wait Time	External Bus Control Normal Mode Wait Time (Default: 111 (~DATA[4])/0000) Specifies the wait cycle count in the first cycle of a Single Cycle or Burst access. Specifies the following wait cycle count when in the Page mode. 0000: 0 wait cycles 0100: 4 wait cycles 1000: 8 wait cycles 1100: 12 wait cycles 0001: 1 wait cycle 0101: 5 wait cycles 1001: 9 wait cycles 1101: 13 wait cycles 0010: 2 wait cycles 0110: 6 wait cycles 1010: 10 wait cycles 1110: 14 wait cycles 0011: 3 wait cycles 0111: 7 wait cycles 1011: 11 wait cycles 1111: 15 wait cycles Specifies a wait cycle count from 0 to 62 that matches PWT when in a mode other than the Page mode. PWT[1:0]: WT[3:0] 000000: 0 wait cycles 010000: 16 wait cycles 110000: 48 wait cycles 000001: 1 wait cycle 010001: 17 wait cycles 110001: 49 wait cycles : : : 001110: 14 wait cycles 011110: 30 wait cycles 111110: 62 wait cycles 001111: 15 wait cycles 011111: 31 wait cycles 111111: External ACK mode Note 1: Value that is the reverse of DATA[4] is set to the LSB of Channel 0 as the default. Note 2: If PWT:WT is set to 0x3f when PM = 00 and RDY = 0, the external bus enters the ACK* Input mode (External ACK mode) without the wait cycle count for the ACK* output being the maximum value. Note 3: WT[0] is used to select Dynamic/Static ACK*/Ready mode when in the Ready mode. Therefore, the Wait cycle count is an even number. Note 4: The WT wait cycle count should be equal to or greater than the PWT Wait cycle count when in the Page mode.	R/W
11:8	CS	Channel Size	External Bus Control Channel Size (Default: 0010/0000) Specifies the channel memory size. 0000: 1 MB 0101: 32 MB *1010: 1 GB 0001: 2 MB 0110: 64 MB 1011-1111: Reserved 0010: 4 MB 0111: 128 MB 0011: 8 MB 1000: 256 MB 0100: 16 MB 1001: 512 MB * The channel memory size can be set up to 512 MB when the memory bus width is 16 bits, or up to 256 MB when the memory bus width is 8 bits. No size larger than this can be set.	R/W

Figure 7.4.1 External Bus Channel Control Register (2/3)

Bit	Mnemonic	Field Name	Description	Read/Write
7	BC	Byte Control	External Bus Byte Control (Default: DATA[5]/0) Specifies whether to use the BWE*[3:0] signal as an asserted Byte Write Enable signal (BWE*[3:0]) only during a Write cycle, or to use it as an asserted Byte Enable signal (BE*[3:0]) that is asserted during both Read and Write cycles. 0: Byte Enable (BE *[3:0]) 1: Byte Write Enable (BWE*[3:0]) Note: DATA[5] is set to Channel 0 as the default.	R/W
6	RDY	Ready Input Mode	External Bus Control Ready Input Mode (Default: 0) Specifies whether to use the Ready mode. 0: Disable the Ready mode. 1: Enable the Ready mode. Note: The Ready mode cannot be used when the Page mode is selected.	R/W
5:4	SP	Bus Speed	External Bus Control Bus Speed (Default: ADDR[7:6] / 00) Specifies the External Bus speed. 00: 1/4 speed (1/4 of the GBUSCLK frequency) 01: 1/3 speed (1/3 of the GBUSCLK frequency) 10: 1/2 speed (1/2 of the GBUSCLK frequency) 11: Full speed (same frequency as GBUSCLK) Note: ADDR[7:6] is set to Channel 0 as the default.	R/W
3	ME	Master Enable	External Bus Control Master Enable (Default: ADDR[8] / 0) Enables a channel. 0: Disable channel 1: Enable channel Note: ADDR[8] is set to Channel 0 as the default.	R/W
2:0	SHWT	Set Up/Hold Wait Time	External Bus Control Setup/Hold Wait Time (Default: 000) Specifies the wait count when switching between the Address and Chip Enable signal, or the Chip Enable Signal and Write Enable/Output Enable signal. * 000: Disable 100: 4 wait cycles 001: 1wait cycle 101: 5 wait cycles 010: 2 wait cycles 110: 6 wait cycles 011: 3 wait cycles 111: 7 wait cycles * Set this bit field to “0” when using it in the Page mode or when performing Burst access.	R/W

Figure 7.4.1 External Bus Channel Control Register (3/3)

7.5 Timing Diagrams

Please take the following points into account when referring to the timing diagrams.

(1) The clock frequency of the SYSCLK signal can be set to one of the following divisions of the internal bus clock (GBUSCLK): 1/1, 1/2, 1/3, or 1/4. Also, the operating reference clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK) for each channel: 1/1, 1/2, 1/3, or 1/4. (See 7.3.8.) The timing diagrams indicate the SYSCLK signal clock frequency and channel operating reference clock frequency as being equivalent.

(2) Both the BWE* signal and BE* signal are indicated in all timing diagrams. The setting of the Channel Control Register (EBCCRn) determines whether the BWE* pin will function as BWE* or BE*.

(3) All Burst cycles in the timing diagrams illustrate examples in which the address increases by increments of 1 starting from 0. However, cases where the CWF (Critical Word First) function of the TX49 core was used or the decrement burst function performed by the DMA Controller was used are exceptions.

(4) The timing diagrams display each clock cycle currently being accessed using the symbols described in the following table.

SWn	Normal Wait Cycles
PWn	Page Wait Cycles
ASn	Set-up Time from SHWT Address Validation to CE Fall
CSn	Set-up Time from SHWT CE Fall to OE/SWE Fall
AHn	Hold Time from SHWT CE Rise to Address Change
CHn	Hold Time from SHWT OE/SWE Rise to CE Rise
ESn	Synch Cycles of the External Input Signal
ACEn	Address Clock Enable Cycles
Sn	Other Cycles

(5) Shaded areas ( ) in the diagrams are undefined values.

7.5.1 ACE* Signal

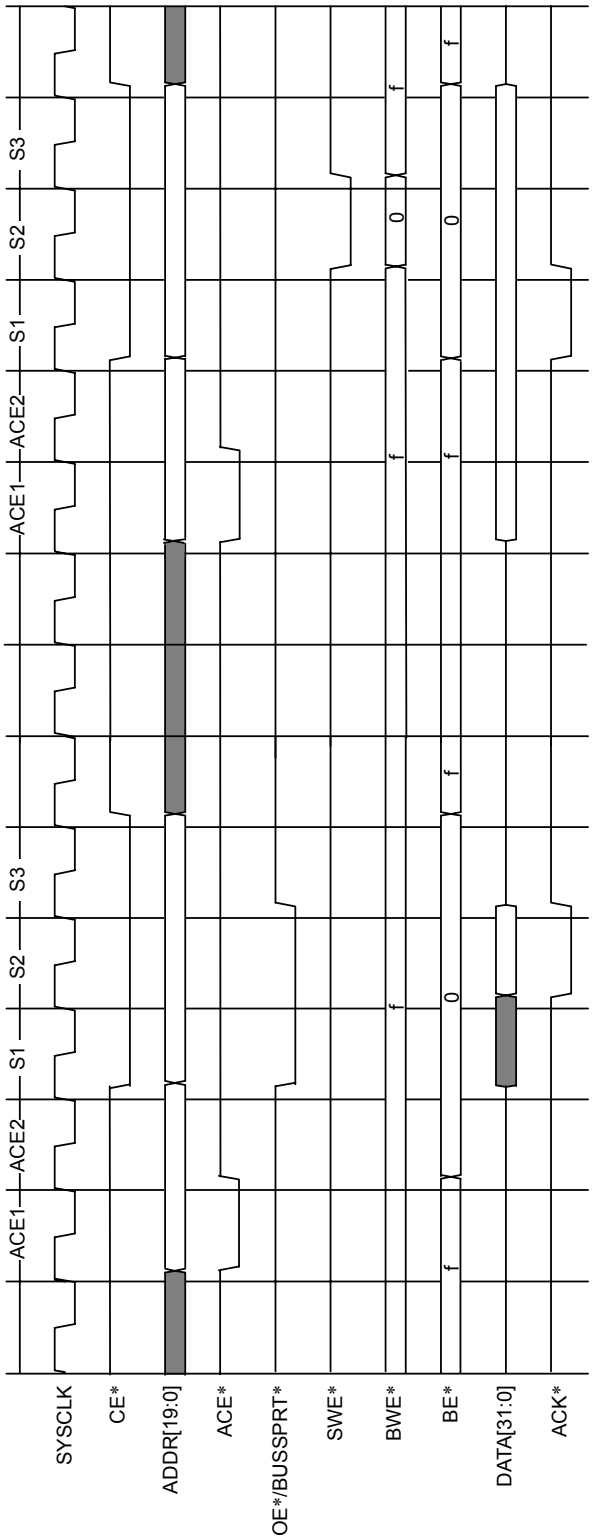


Figure 7.5.1 ACE* Signal (CCFG.ACEHOLD=1, PWT: WT=0, SHWT=0, Normal)

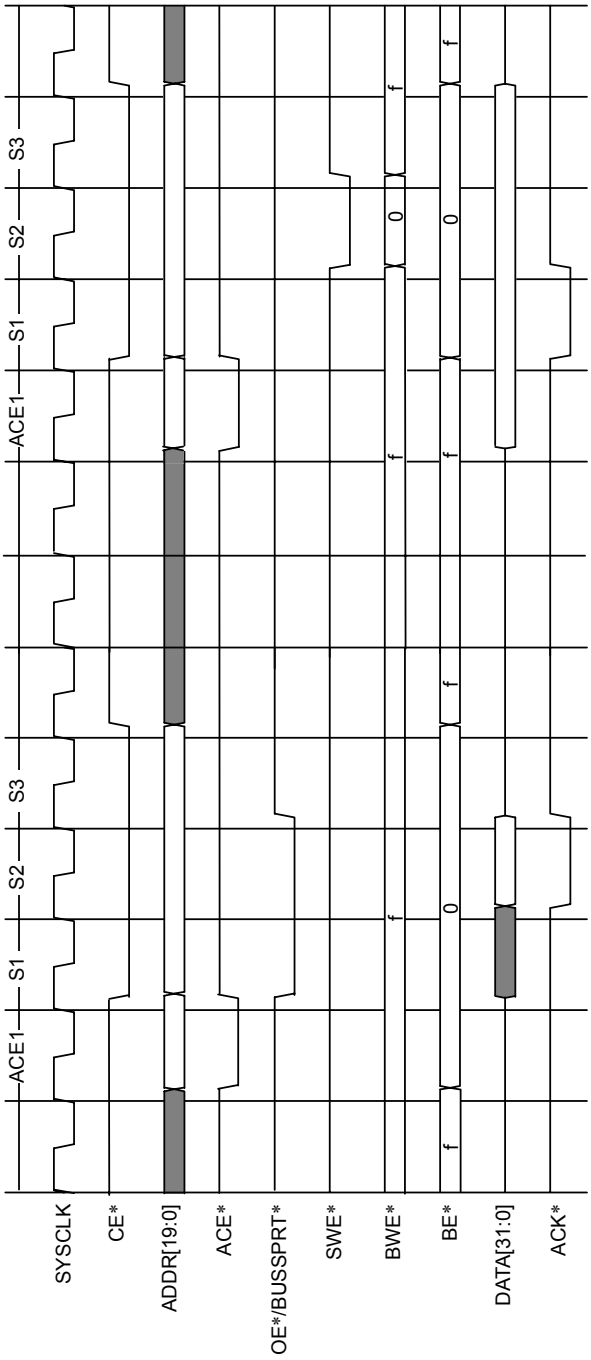


Figure 7.5.2 ACE* Signal (CCFG.ACEHOLD=0, PWT: WT=0, SHWT=0, Normal)

7.5.2 Normal mode access (Single, 32-bit Bus)

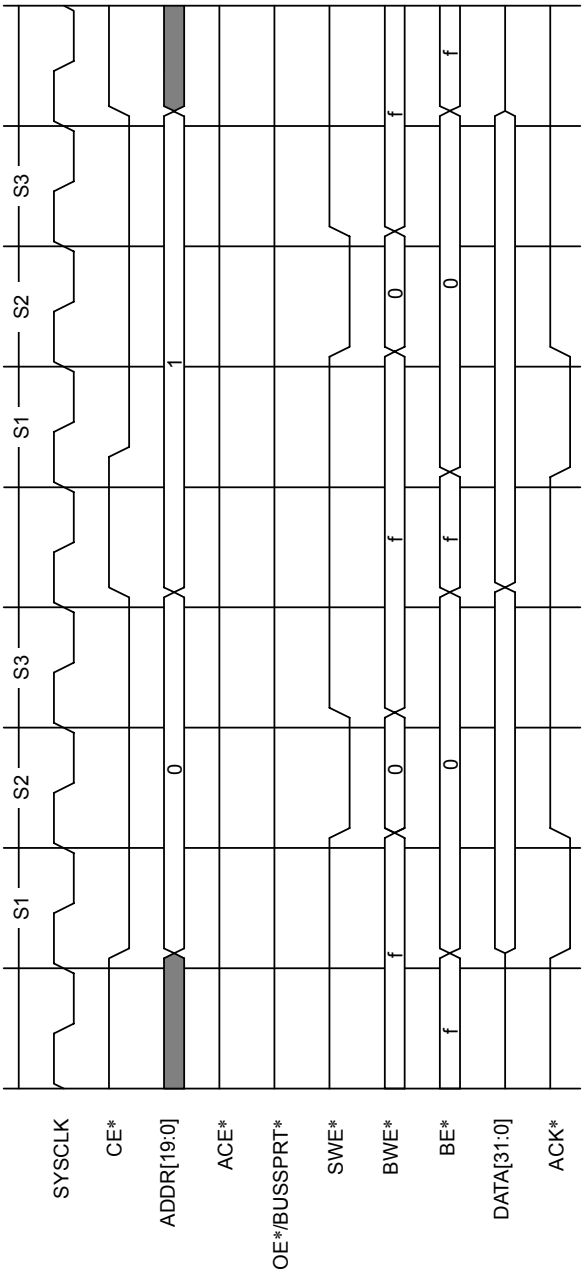


Figure 7.5.3 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)

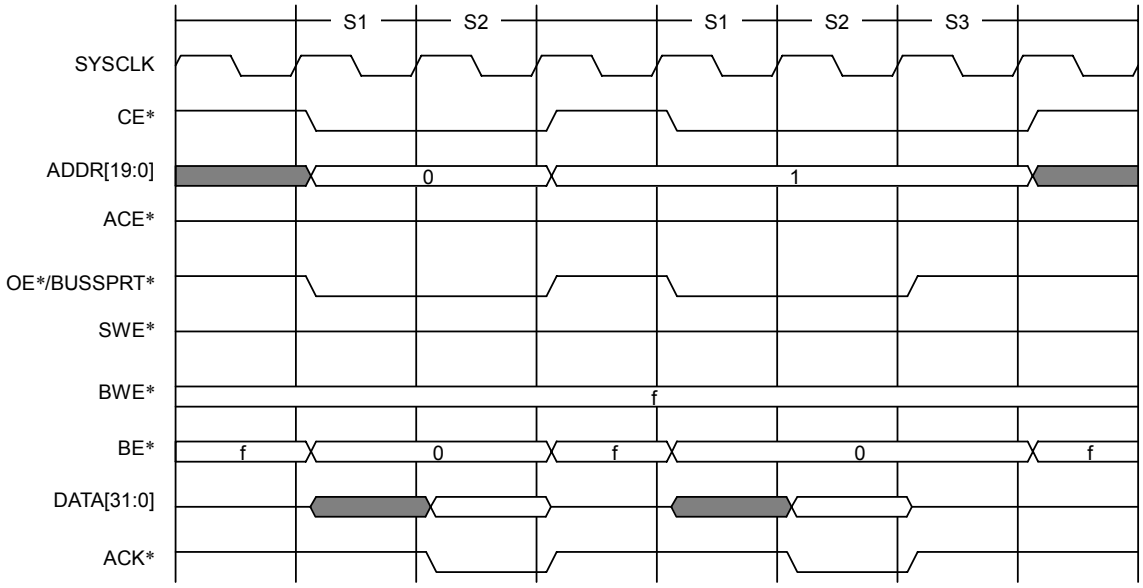


Figure 7.5.4 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)

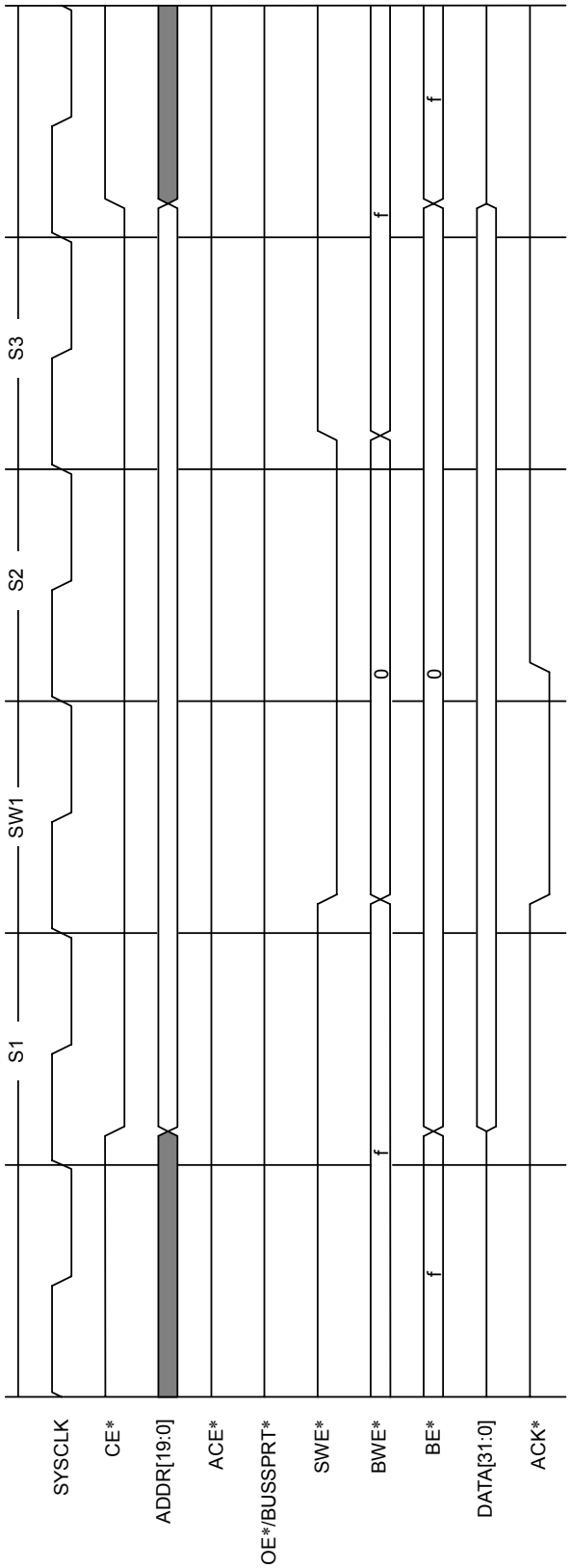


Figure 7.5.5 1-word Single Write (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)

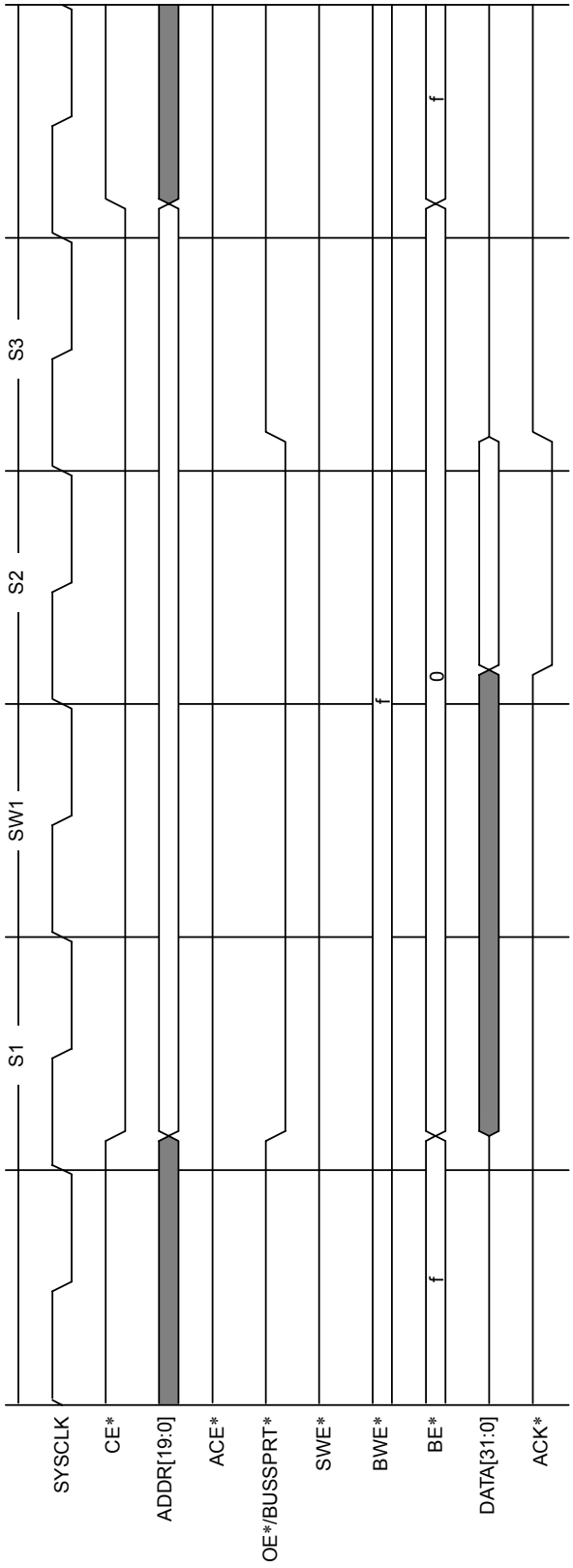


Figure 7.5.6 1-word Single Read (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)

7.5.3 Normal mode access (Burst, 32-bit Bus)

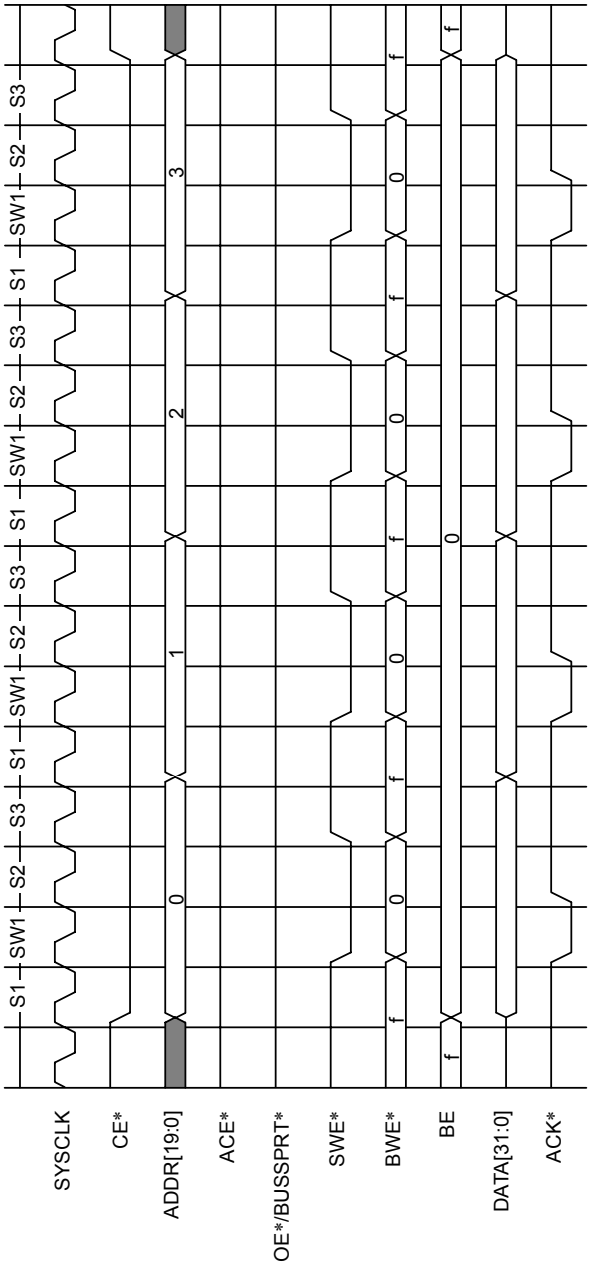


Figure 7.5.7 4-word Burst Write (PWT: WT=1, SHWT=0, Normal, 32-bit Bus)

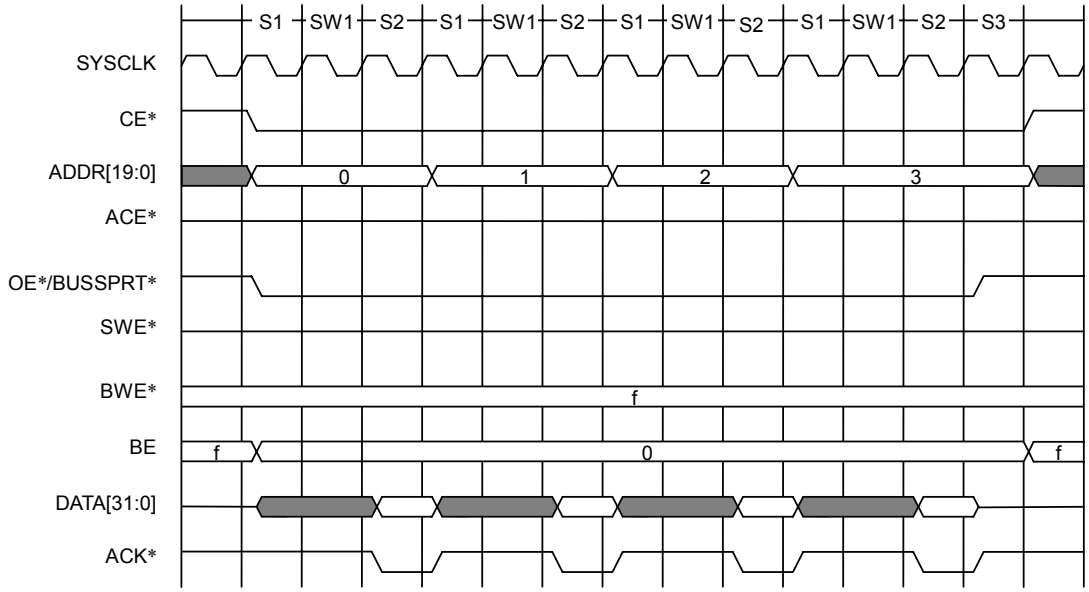


Figure 7.5.8 4-word Burst Write (PWT: WT=1, SHWT=0, Normal, 32-bit Bus)

7.5.4 Normal Mode Access (Single, 16-bit bus)

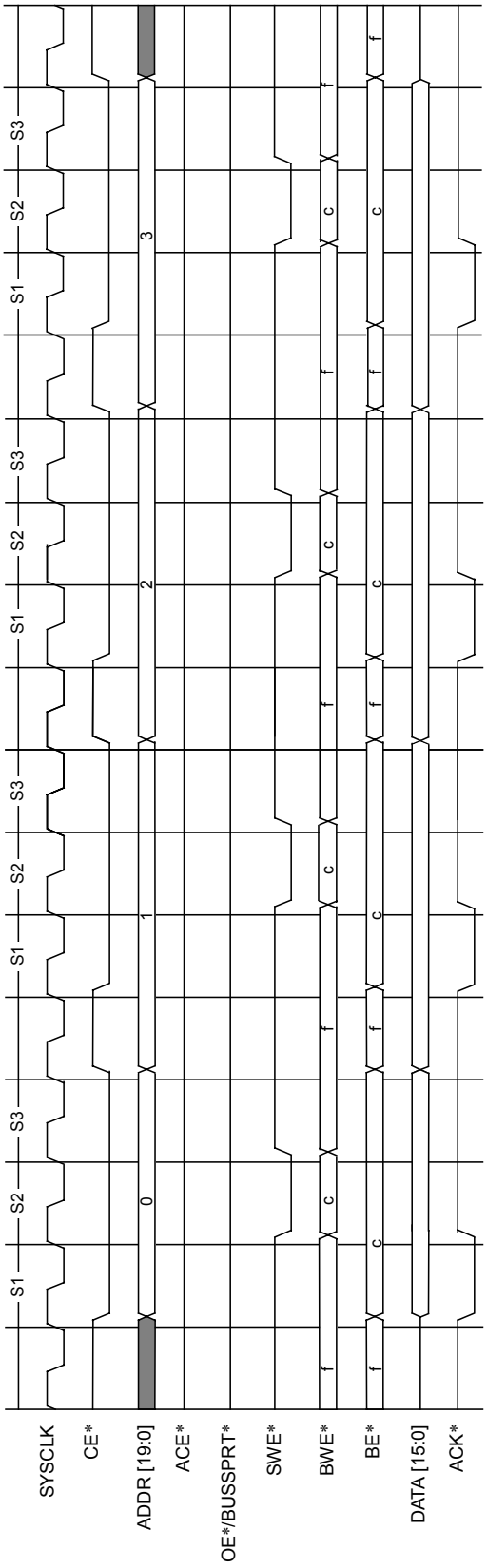


Figure 7.5.9 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

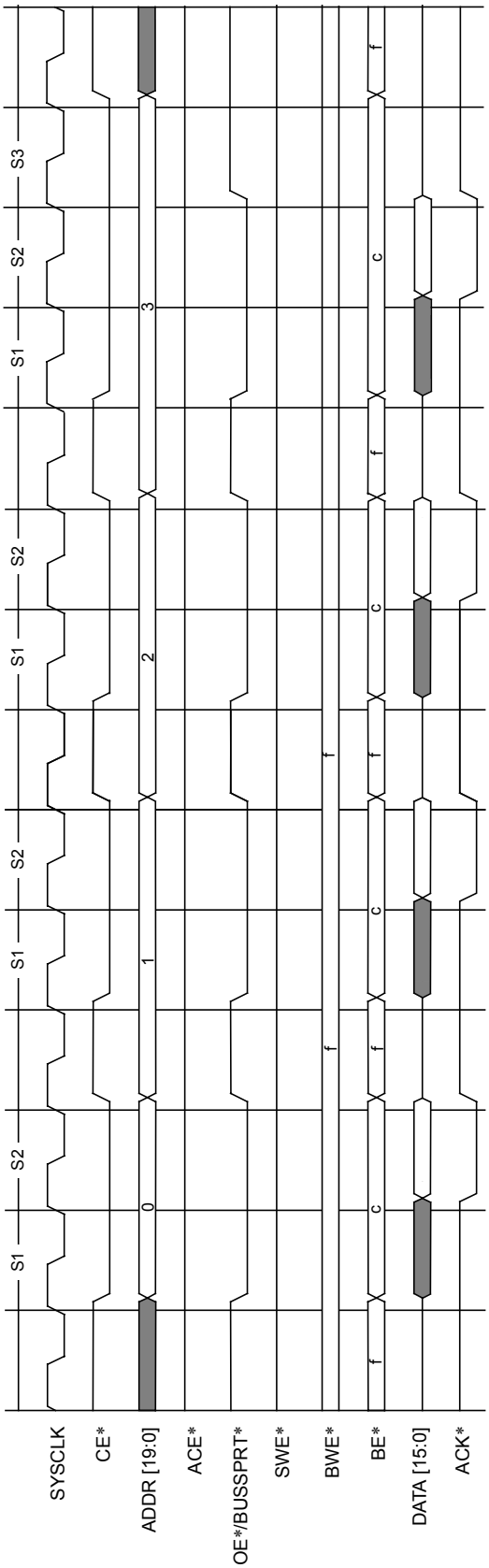


Figure 7.5.10 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

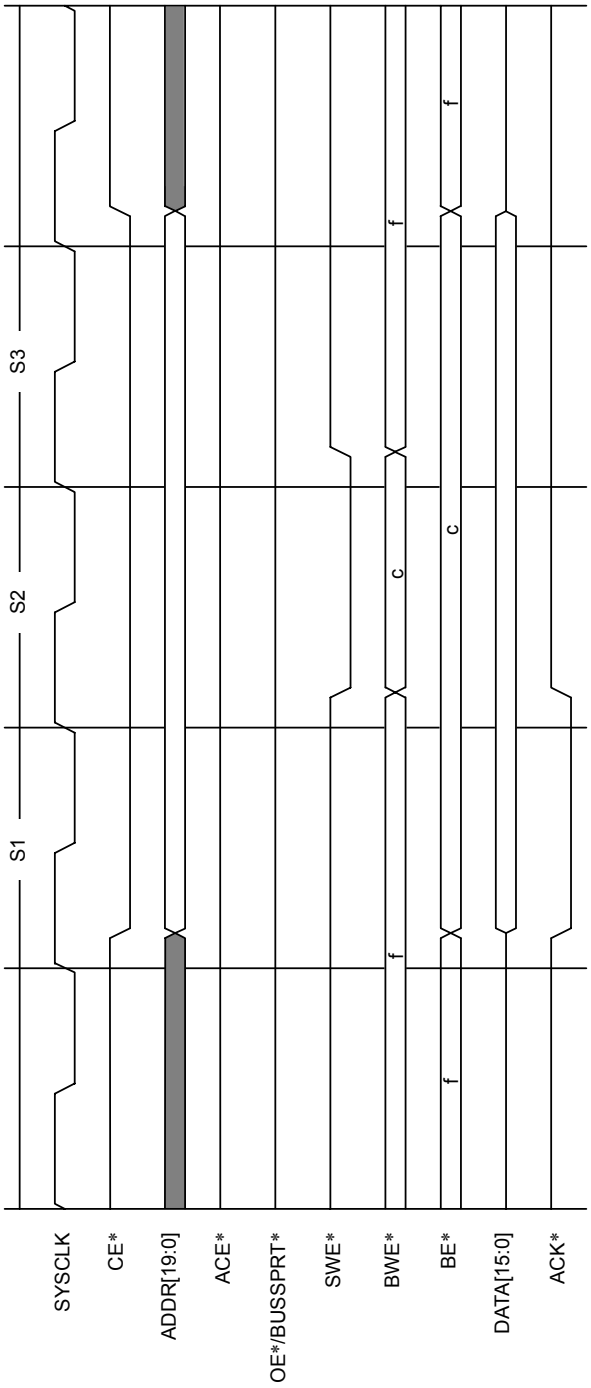


Figure 7.5.11 Half-word Single Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

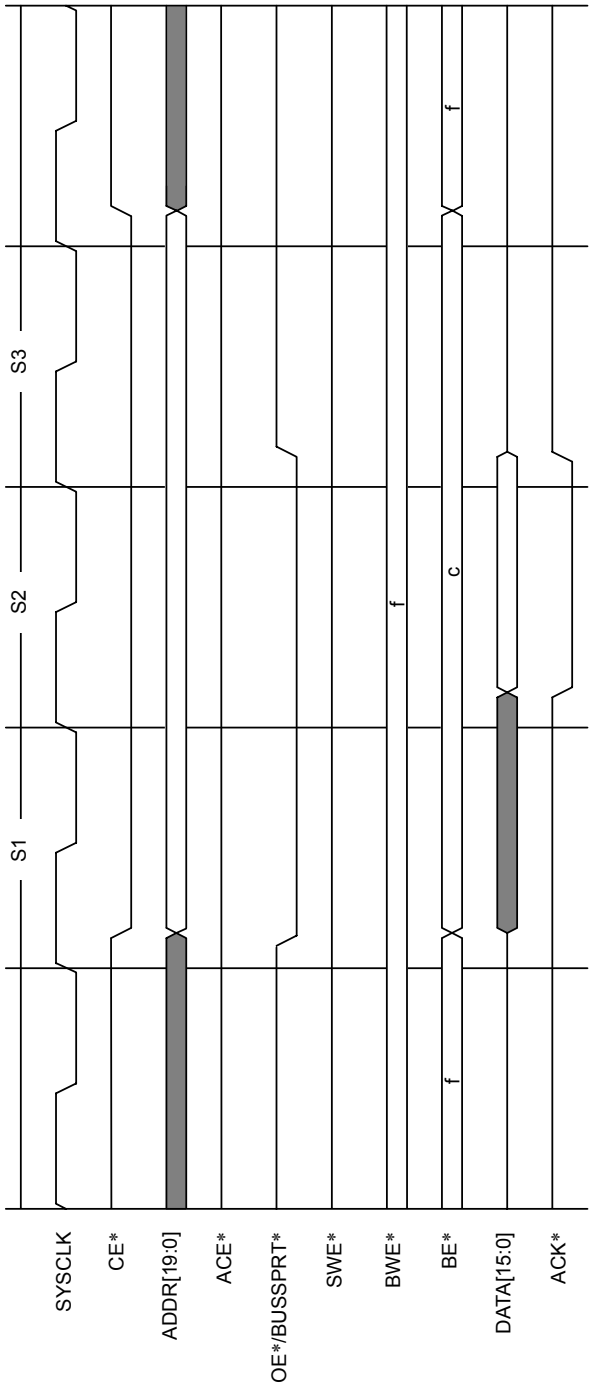


Figure 7.5.12 Half-word Single Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

7.5.5 Normal Mode Access (Burst, 16-bit Bus)

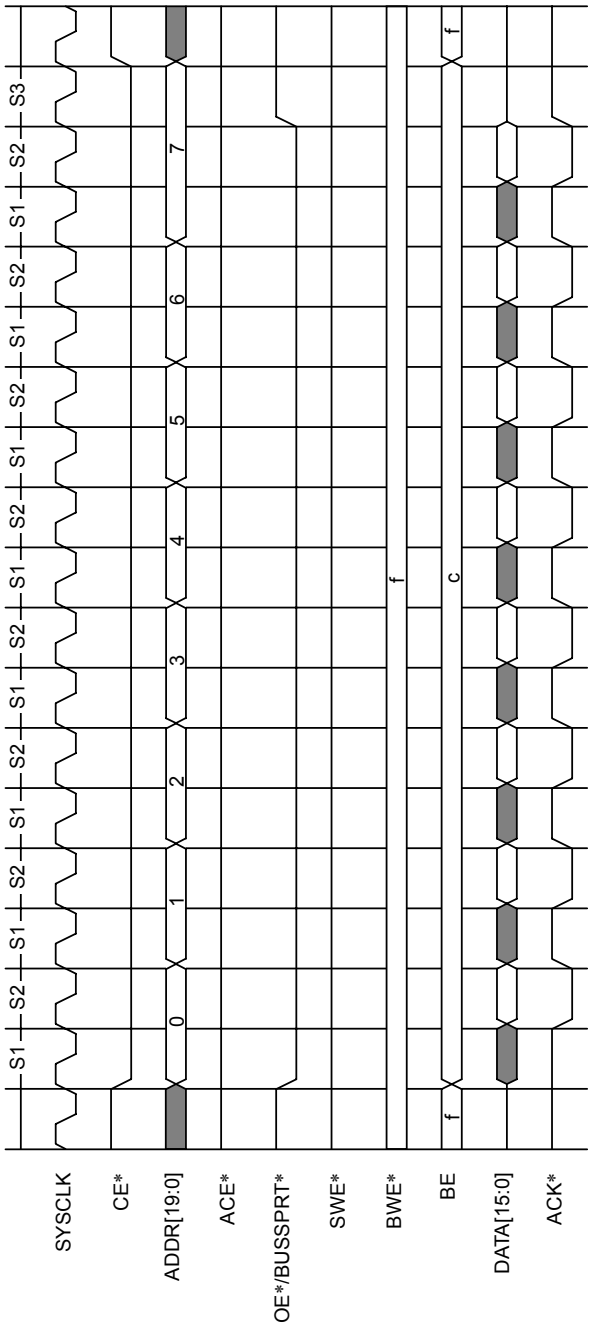


Figure 7.5.13 4-word Burst Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

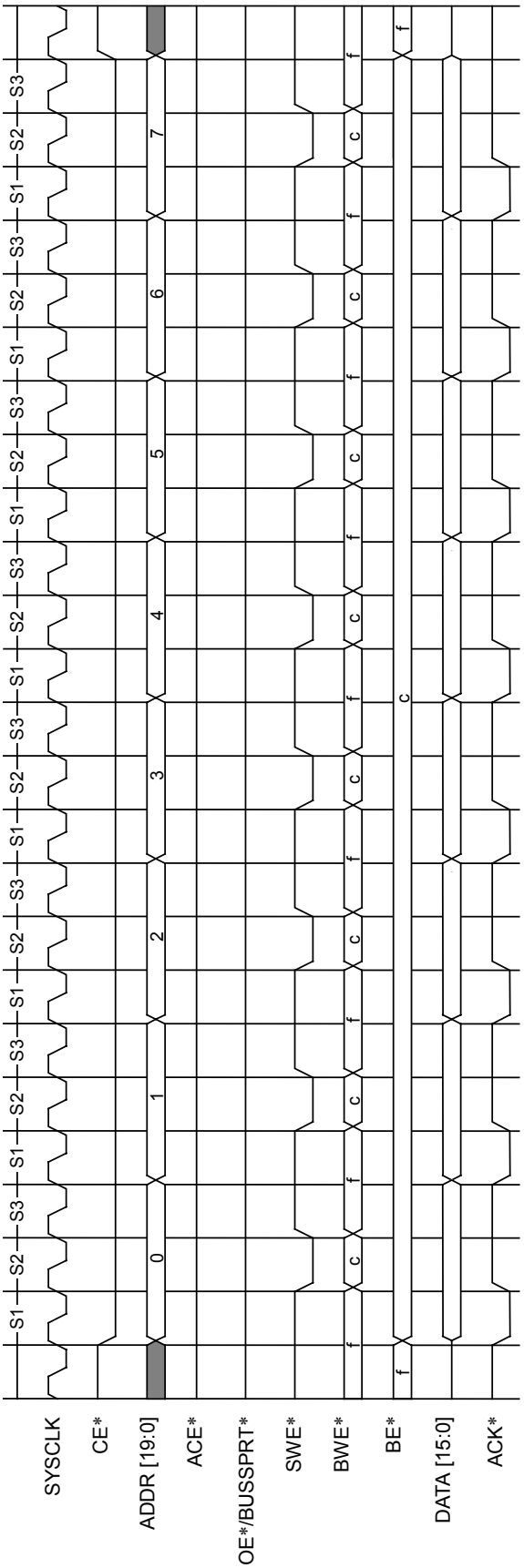


Figure 7.5.14 4-word Burst Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

7.5.6 Normal Mode Access (Single, 8-bit Bus)

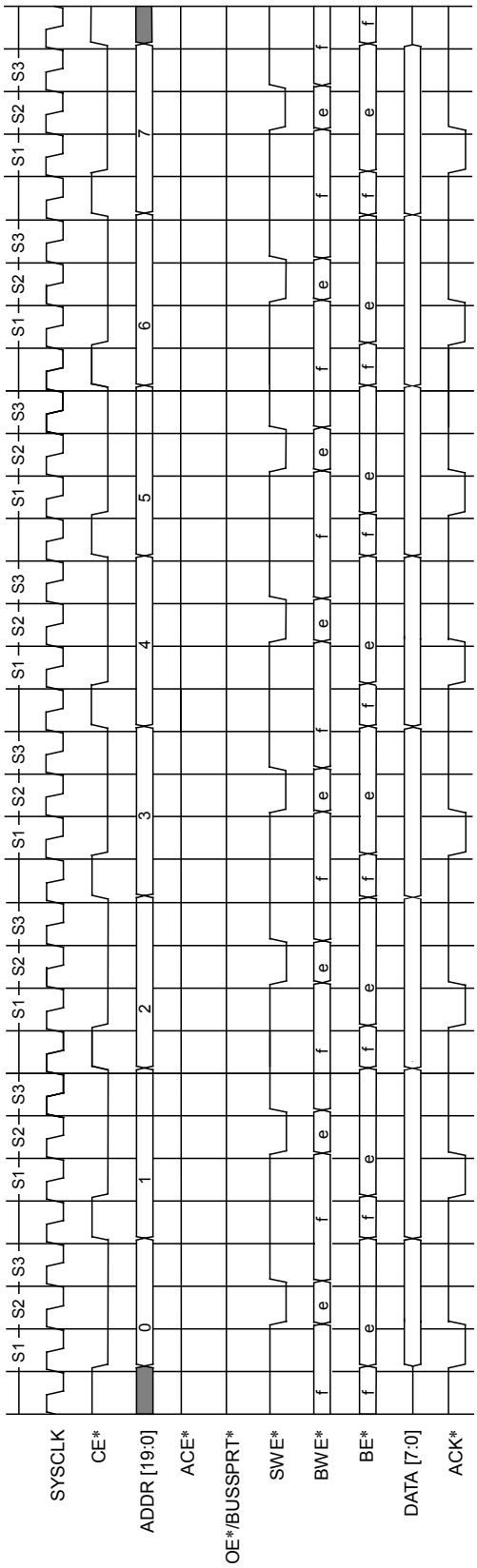


Figure 7.5.15 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

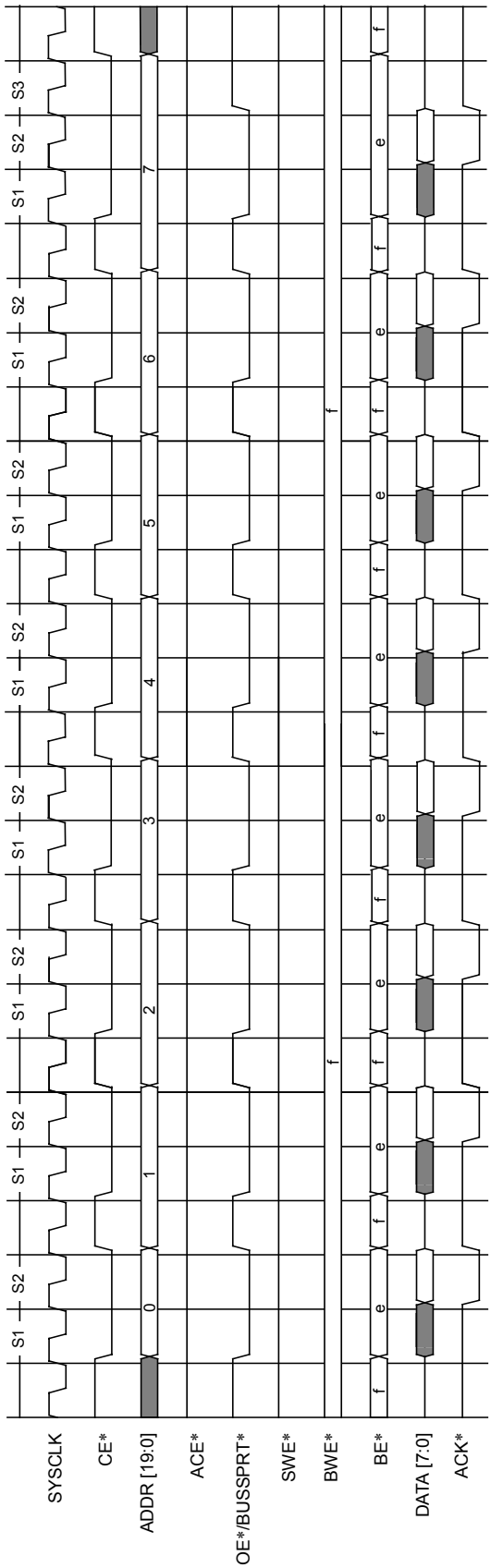


Figure 7.5.16 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

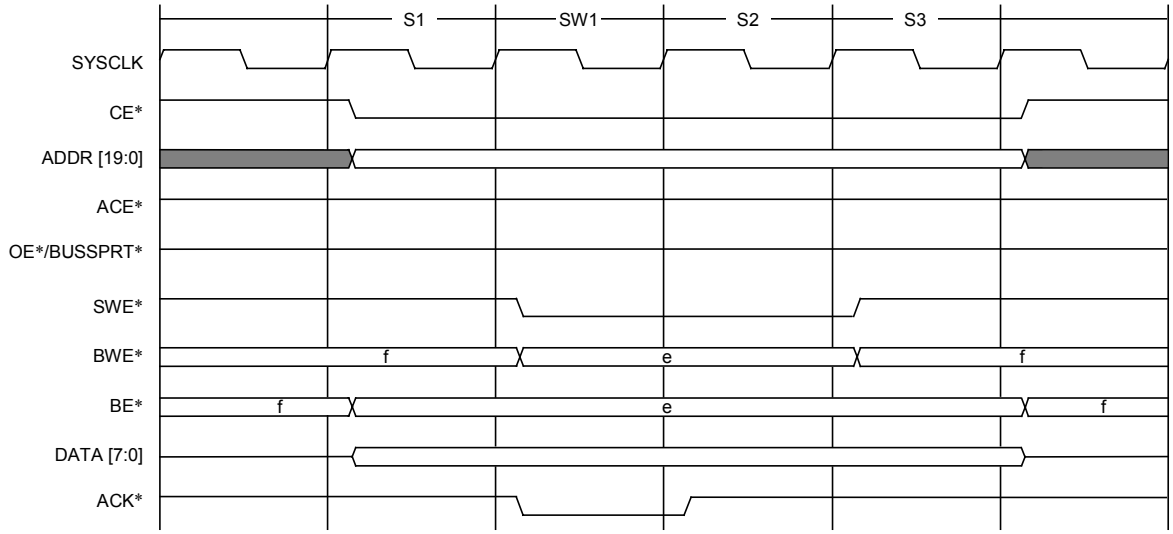


Figure 7.5.17 1-byte Single Write (PWT: WT=1, SHWT=0, Normal, 8-bit Bus)

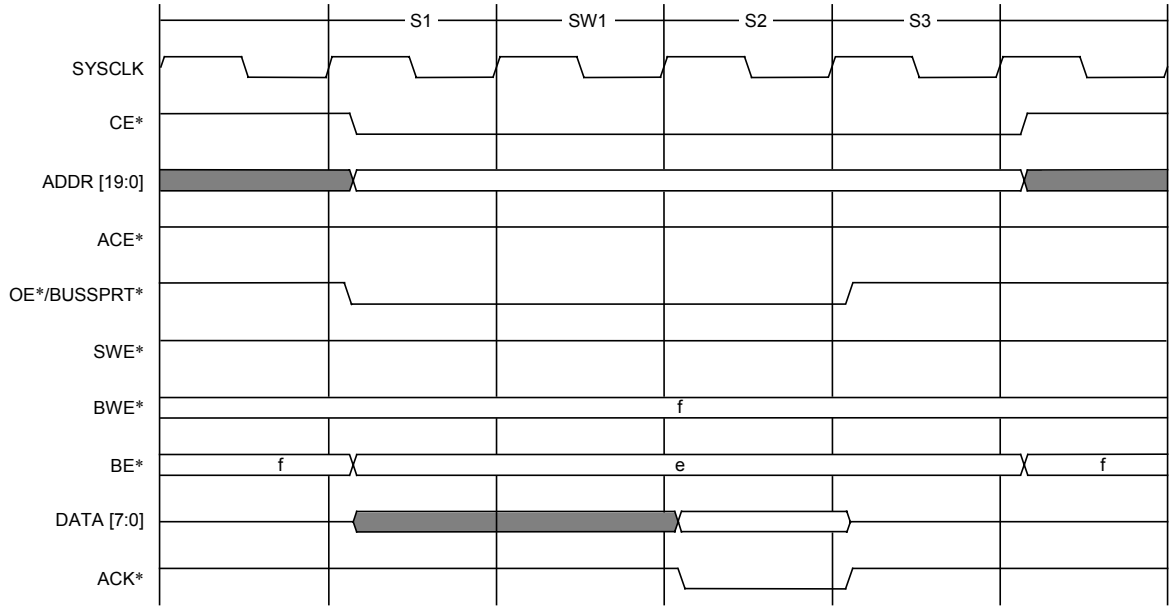


Figure 7.5.18 1-byte Single Read (PWT: WT=1, SHWT=0, Normal, 8-bit Bus)

7.5.7 Normal Mode Access (Burst, 8-bit Bus)

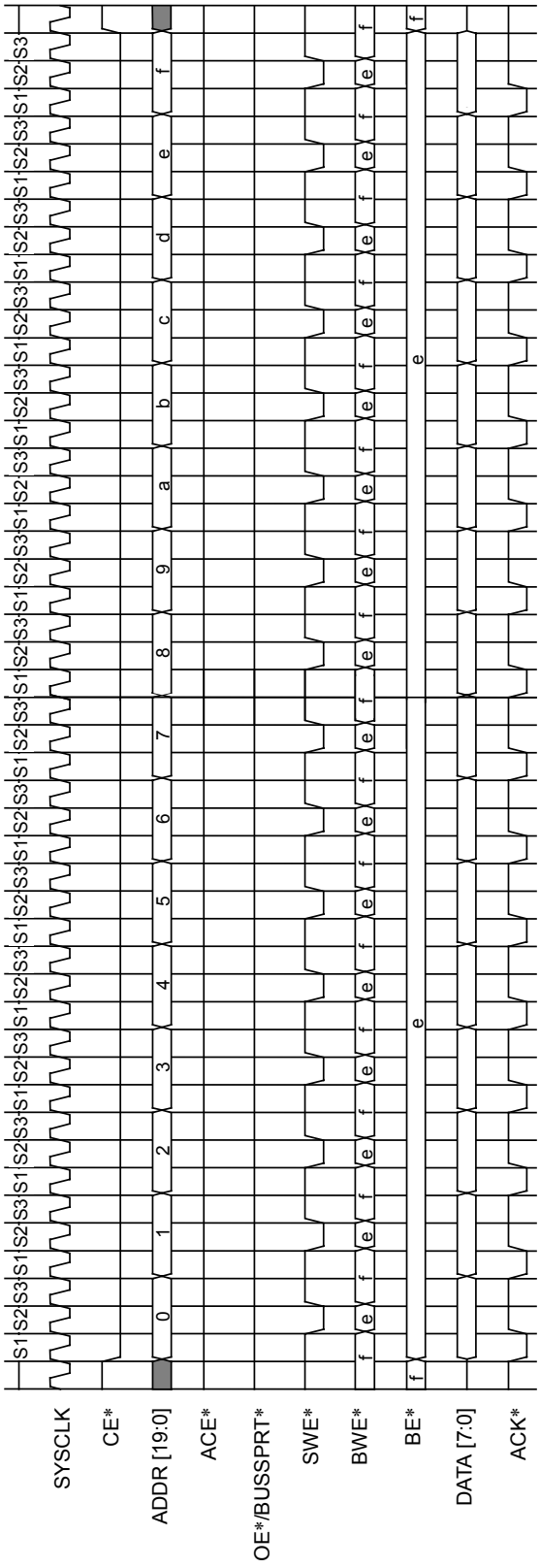
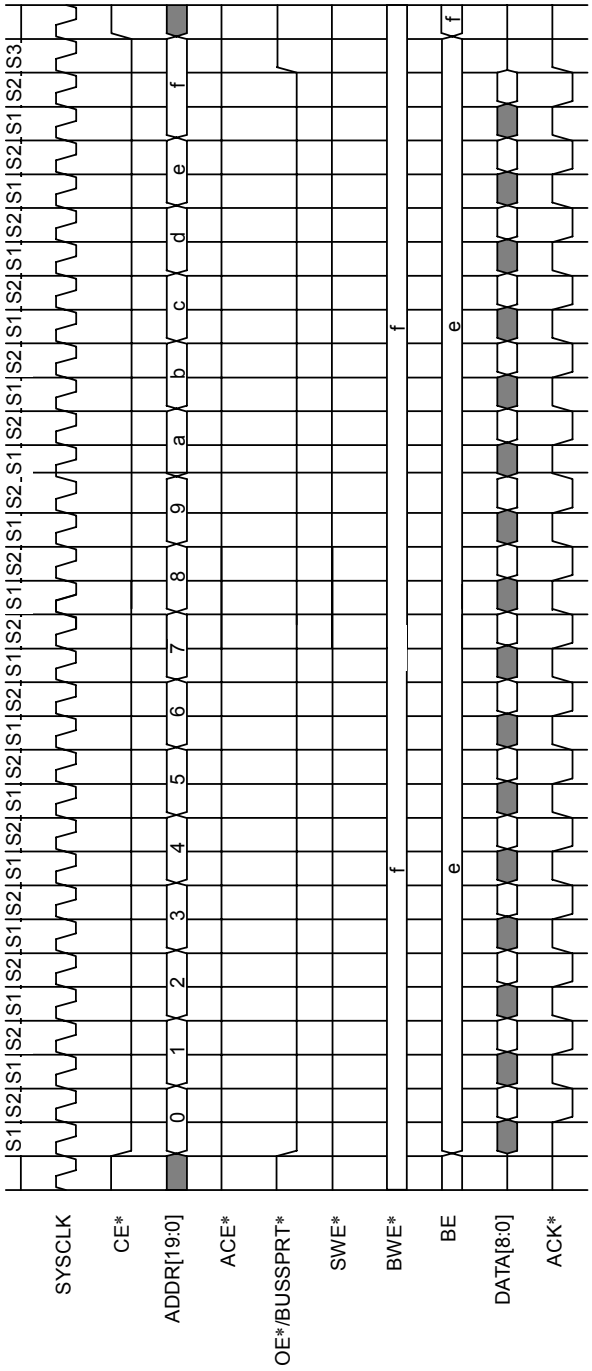


Figure 7.5.19 4-word Burst Write (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)


SYSCLK
CE*
ADDR[19:0]
ACE*
OE*/BUSSPRT*
SWE*
BWE*
BE
DATA[8:0]
ACK*
S1 S2 S1 S2 S1 S2

7.5.8 Page Mode Access (Burst, 32-bit Bus)

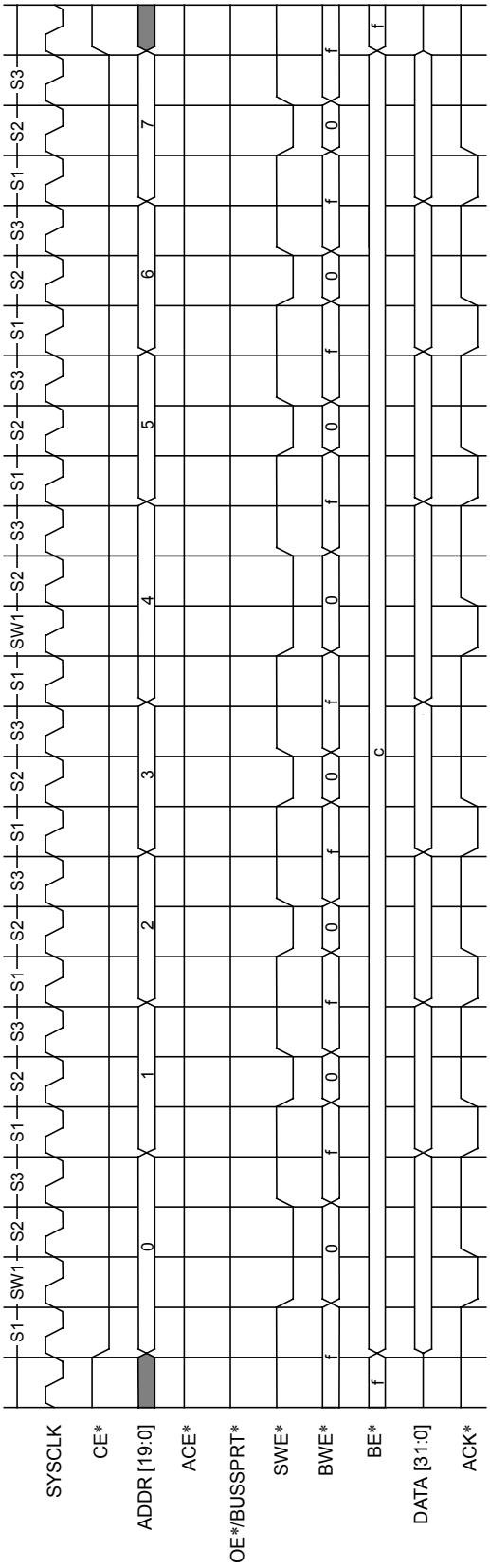


Figure 7.5.21 8-word Burst Write (WT=1, PWT=0, SHWT=0, 4-page, 32-bit Bus)

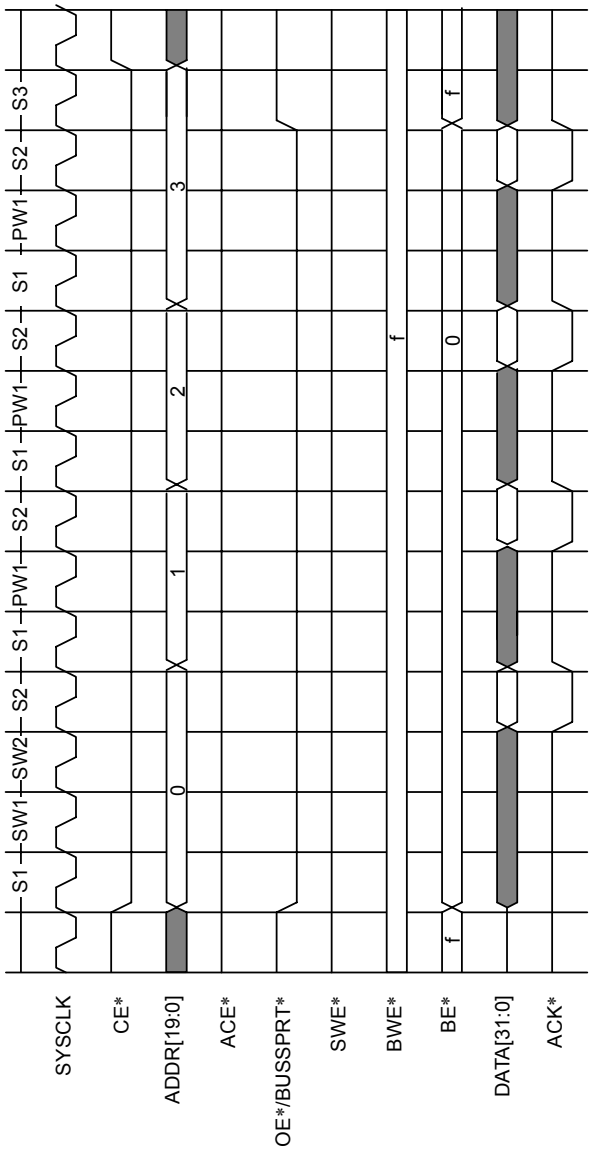
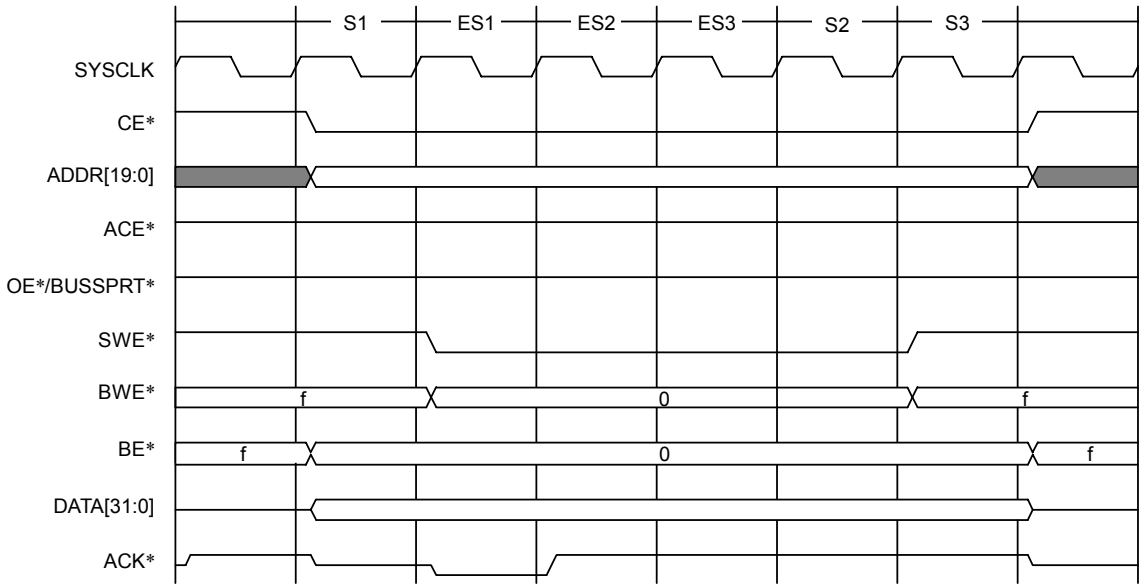


Figure 7.5.22 4-word Burst Read (WT=2, PWT=1, SHWT=0, 4-page, 32-bit Bus)

7.5.9 External ACK Mode Access (32-bit Bus)



Note 1: The TX4938 sets the ACK* signal to High Impedance in the S1 State.

Note 2: External devices drive the ACK* signal to Low (assert the signal) by the ES1 State.

Note 3: External devices drive the ACK* signal to High (deassert the signal) in the ES2 State. If an external device is late in asserting ACK*, then the Wait State is inserted for the amount of time the external device is late. If a certain condition is met, it is okay for the ACK* signal to be driven to Low for 1 clock cycle or more. See 7.3.7.4 ACK* Input Timing (External ACK Mode) for more information.

Figure 7.5.23 1-word Single Write (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

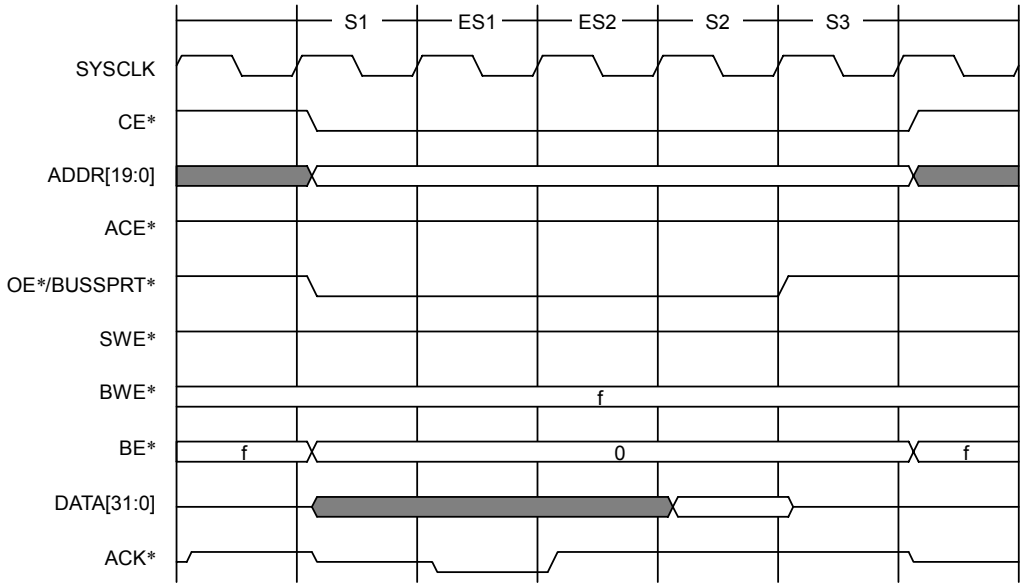


Figure 7.5.24 1-word Single Read (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

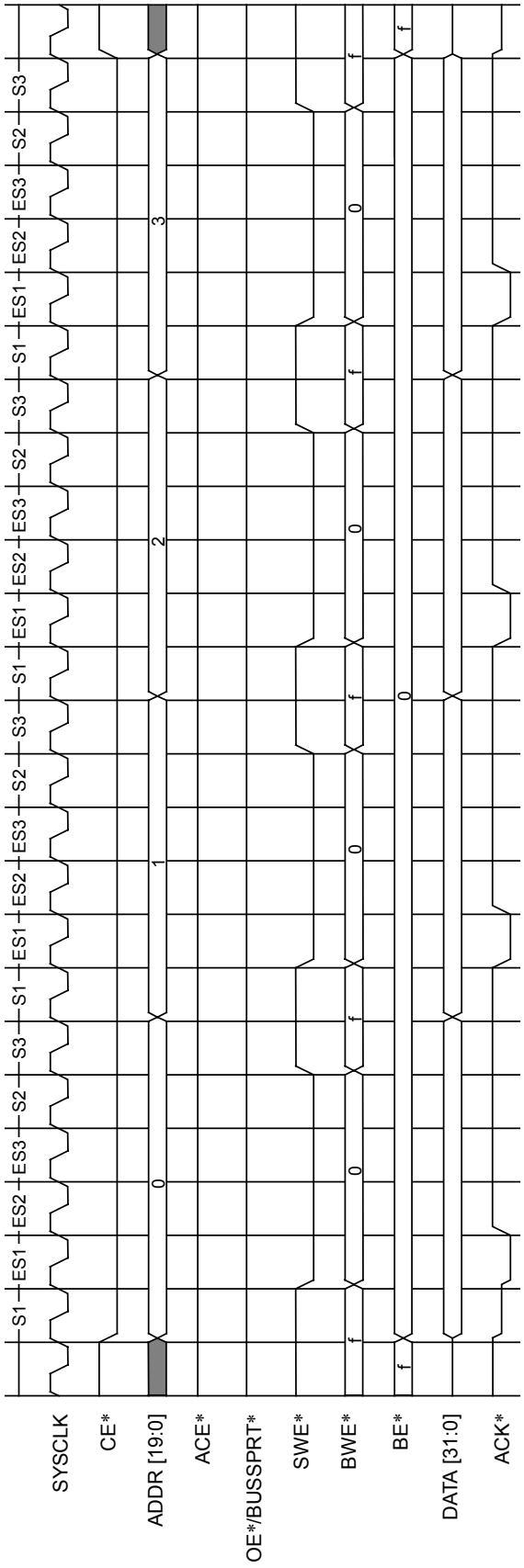


Figure 7.5.25 4-word Burst Write (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

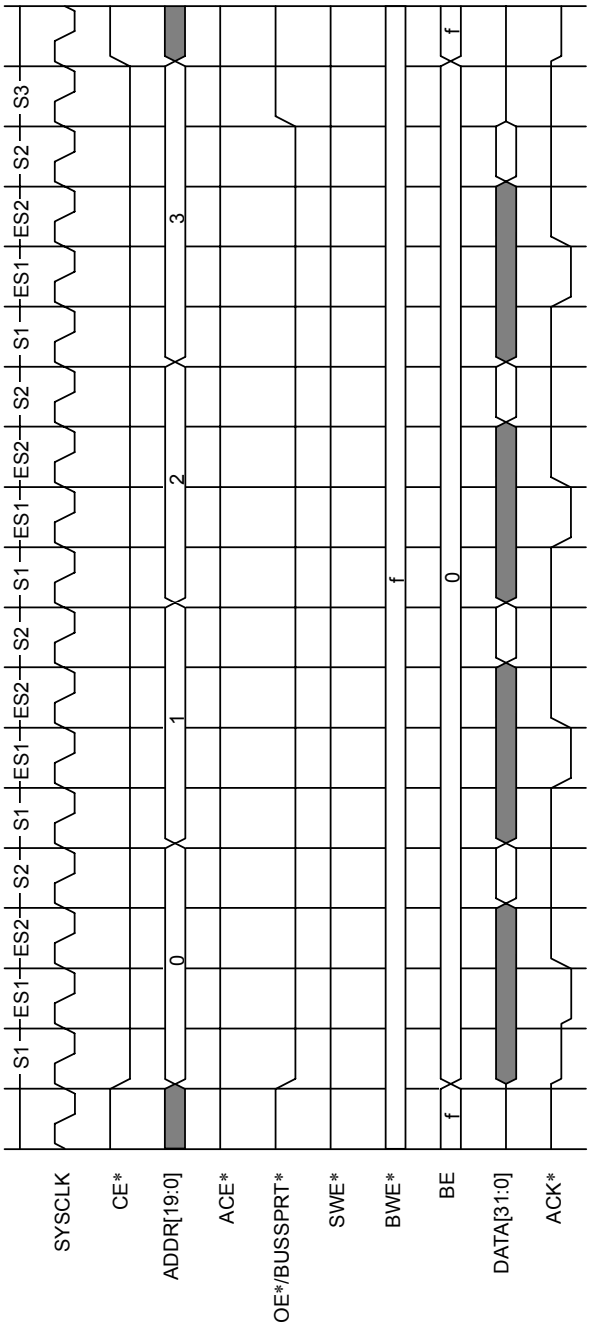


Figure 7.5.26 4-word Burst Read (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

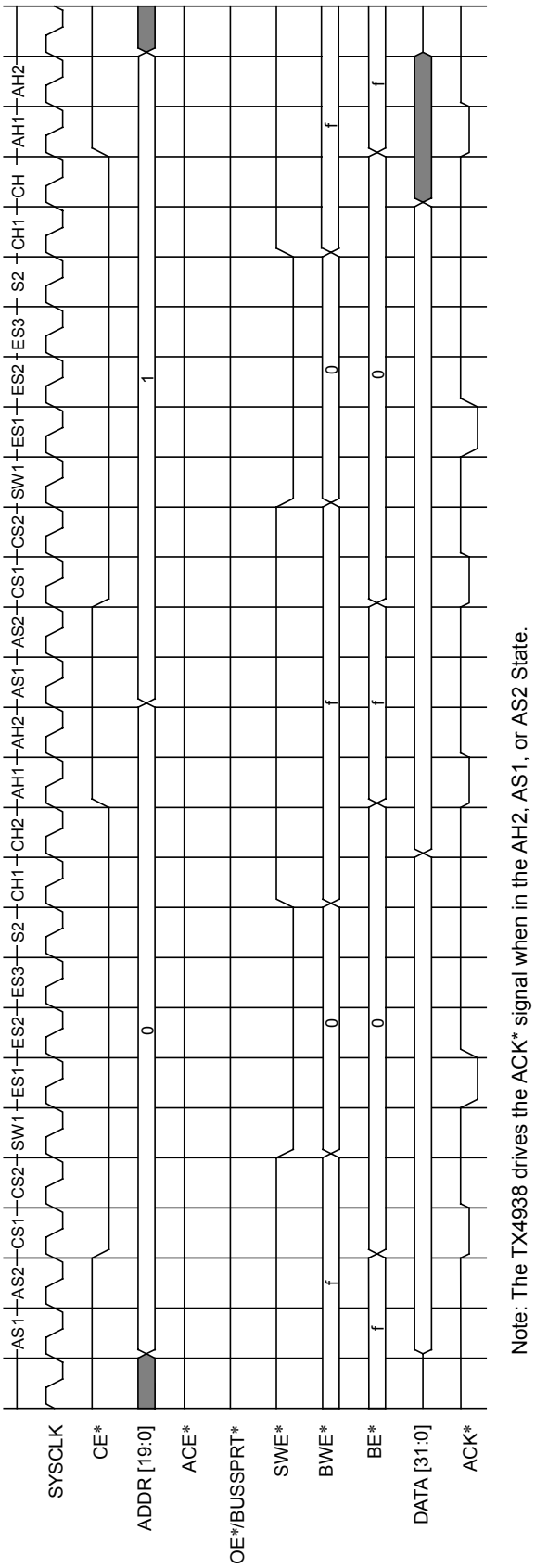


Figure 7.5.27 Double-word Single Write (1 Wait, SHWT=2, External ACK*, 32-bit Bus)

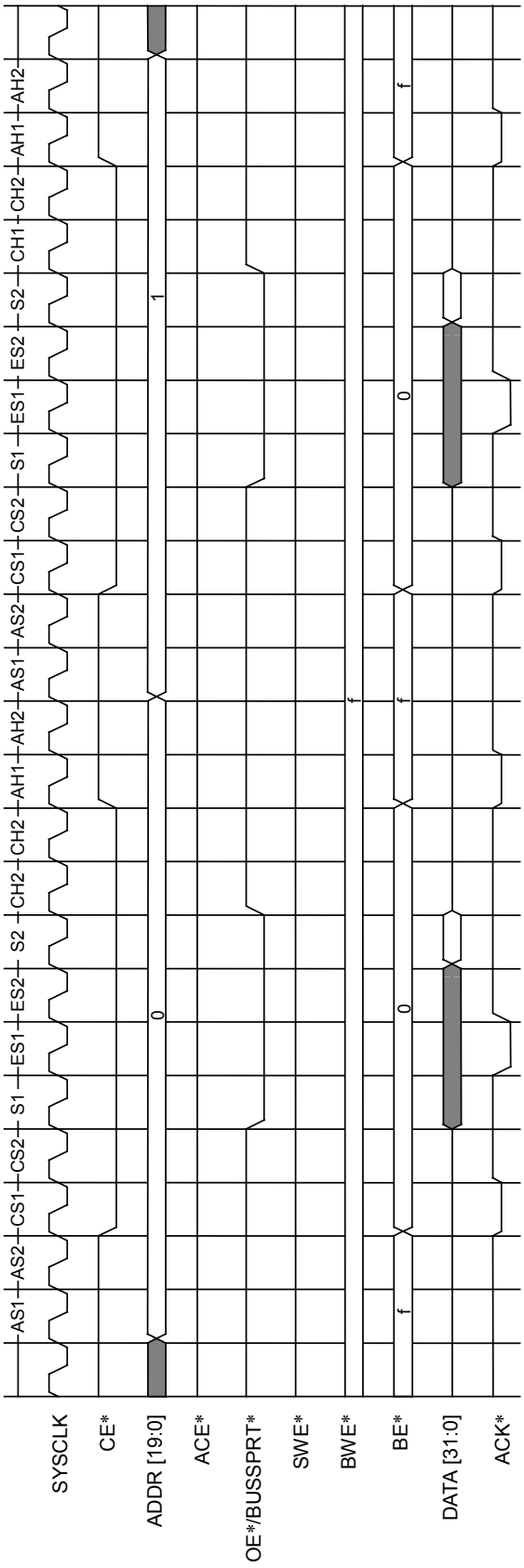


Figure 7.5.28 Double-word Single Read (0 Wait, SHWT=2, External ACK*, 32-bit Bus)

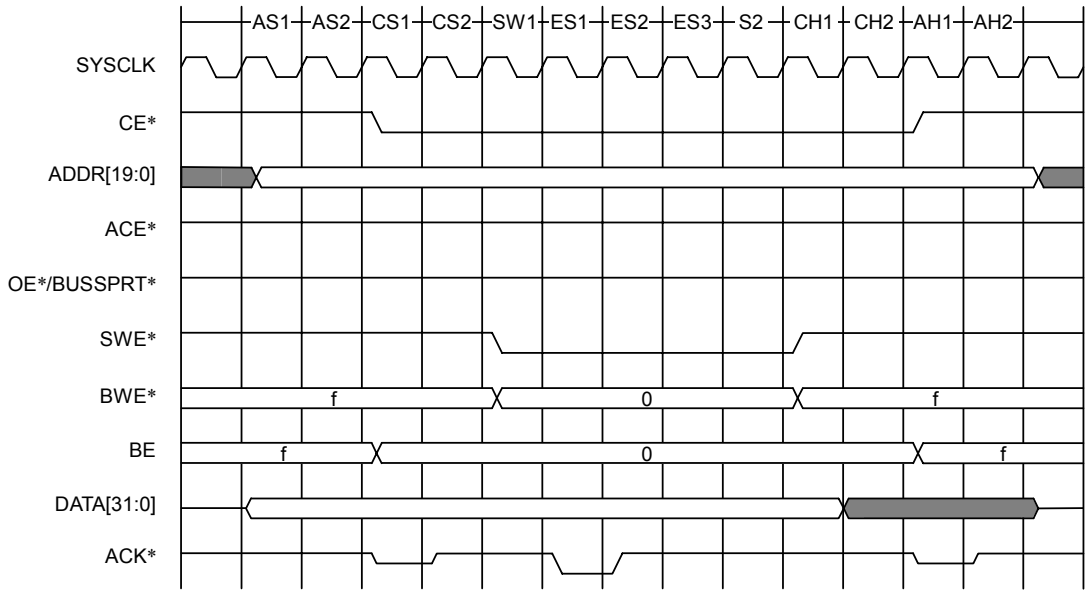


Figure 7.5.29 1-word Single Write (1 Wait, SHWT=2, External ACK*, 32-bit Bus)

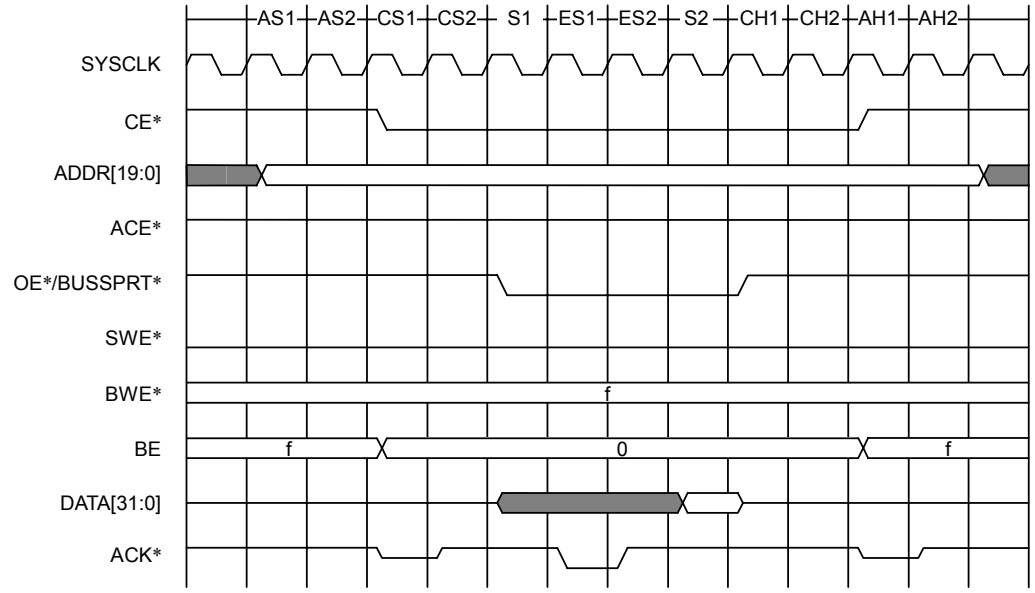


Figure 7.5.30 1-word Single Read (0 Wait, SHWT=2, External ACK*, 32-bit Bus)

7.5.10 READY Mode Access (32-bit Bus)

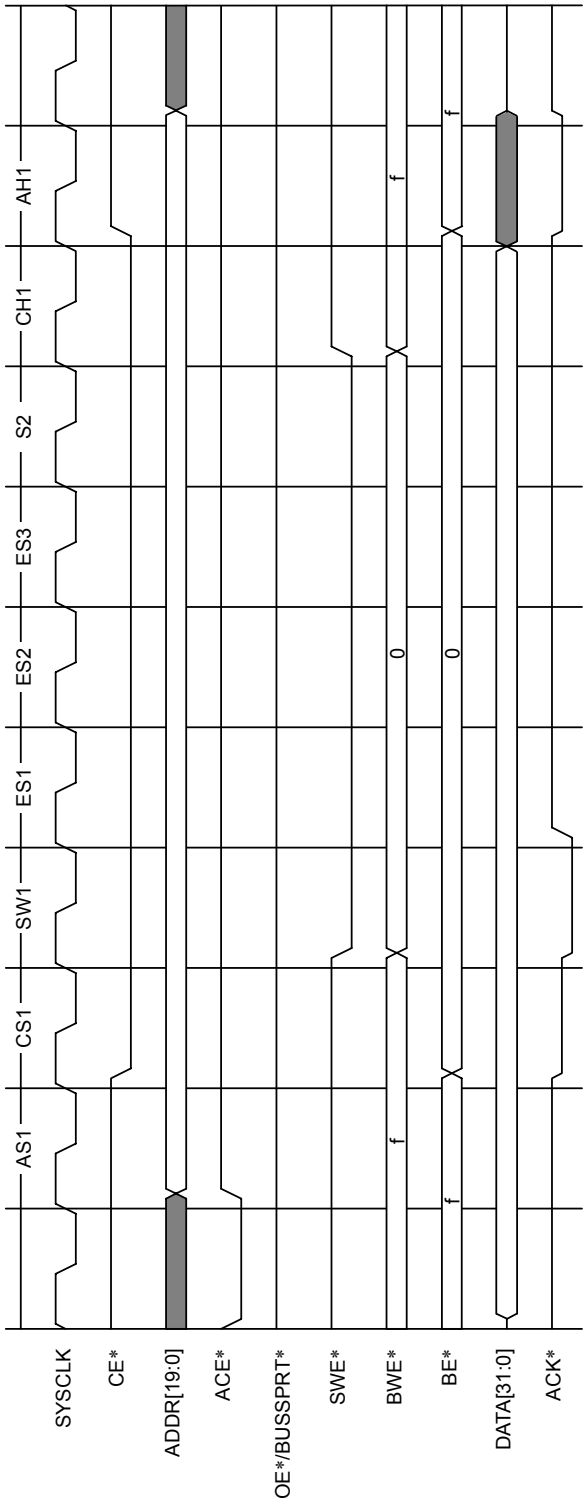


Figure 7.5.31 1-word Single Write (PWT: WT=2, SHWT=1, READY, 32-bit Bus)

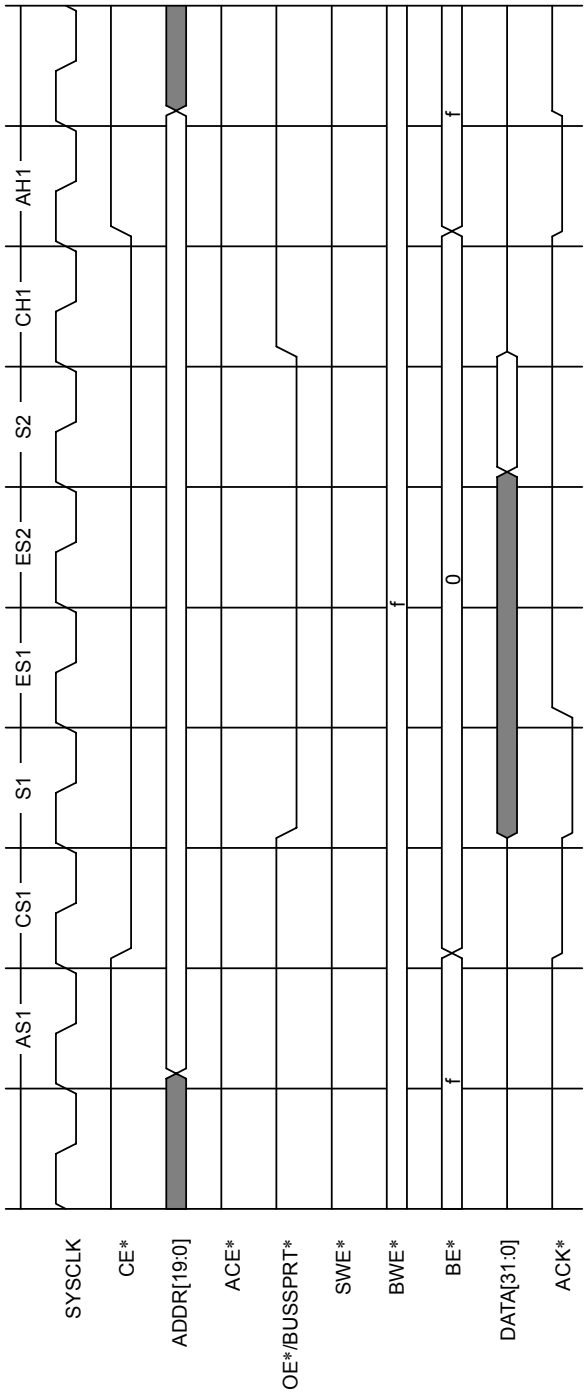


Figure 7.5.32 1-word Single Read (PWT: WT=2, SHWT=1, READY, 32-bit Bus)

7.5.11 ISA IO Space Access

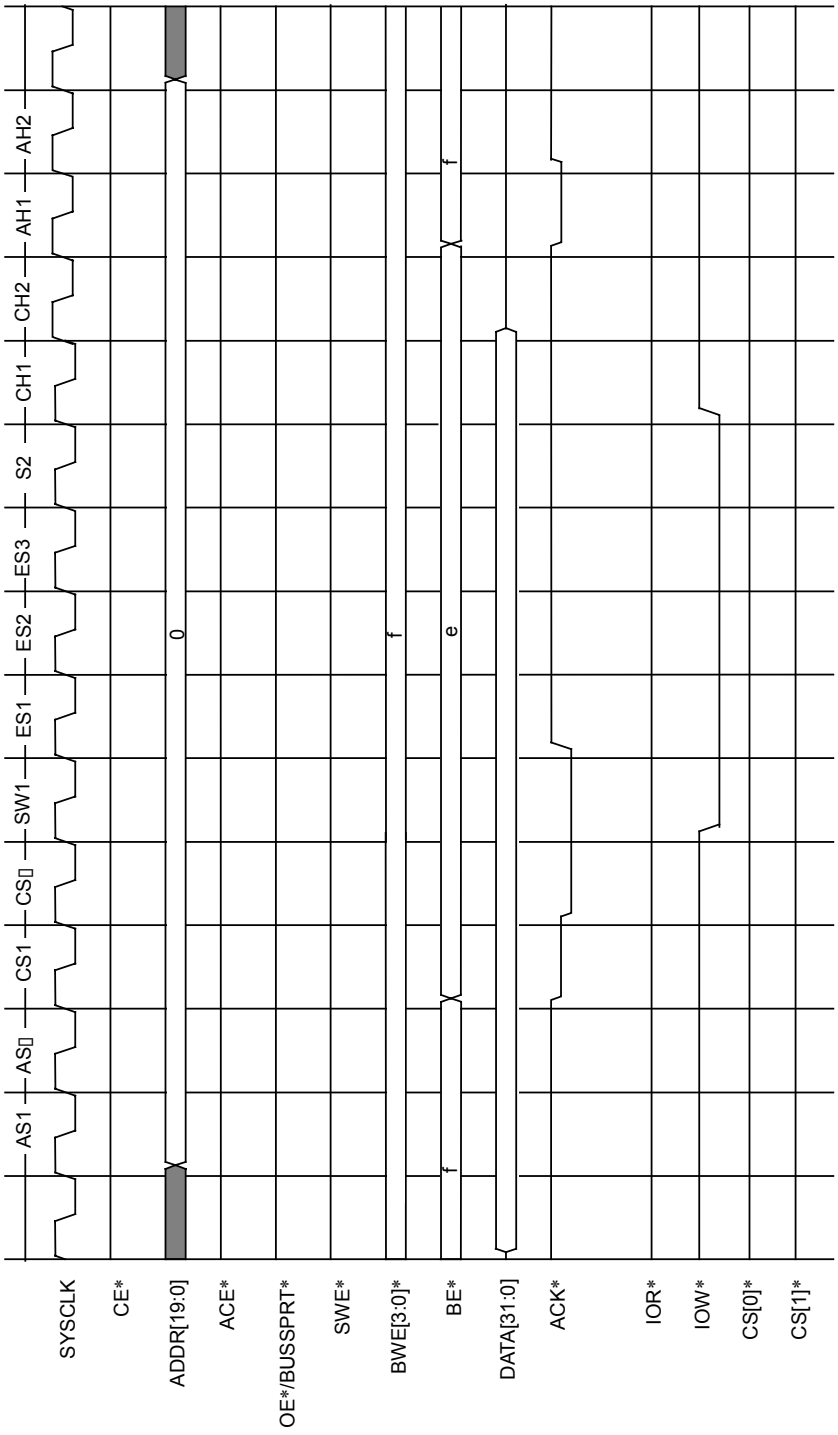


Figure 7.5.33 1-byte Write Access for the ISA IO Space (READY, SHWT=2, 8-bit Bus)

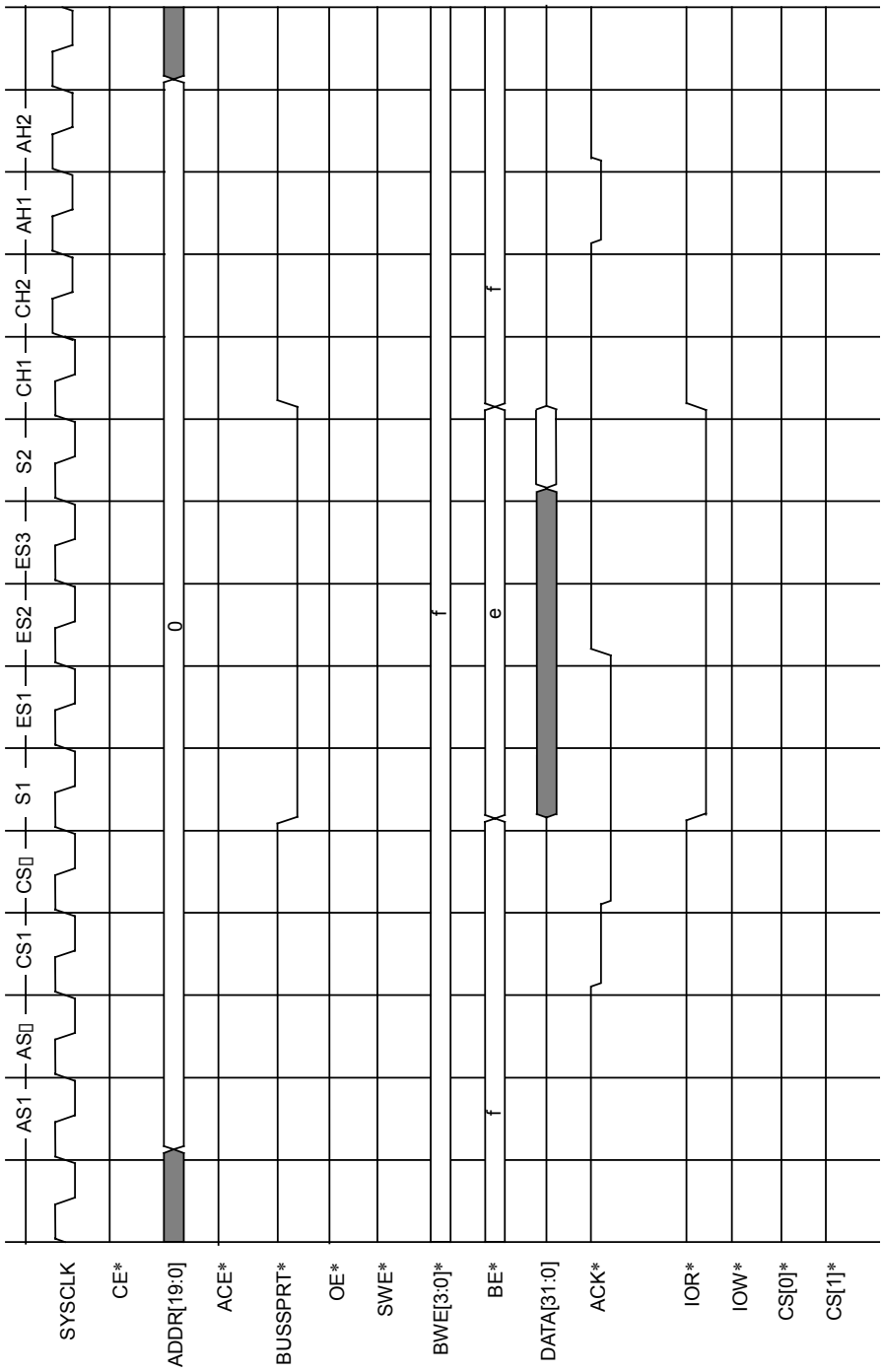


Figure 7.5.34 1-byte Read Access for the ISA IO Space (READY, SHWT=2, 8-bit Bus)

7.5.12 ATA/PIO Transfer Mode Access

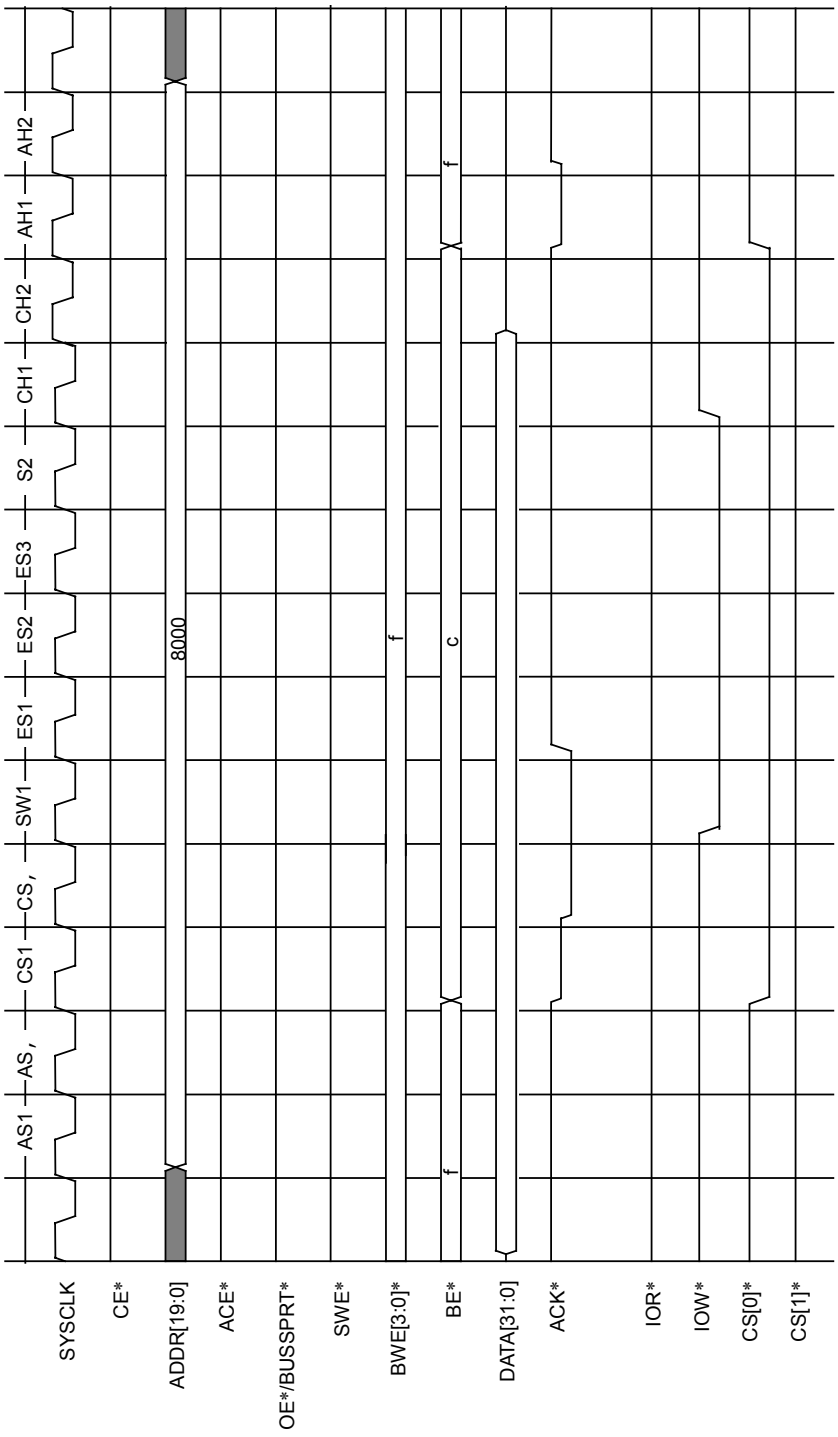


Figure 7.5.35 1-half-word Write Access for the ATA/PIO Space (CS0) (READY, SHWT=2, 16-bit Bus)

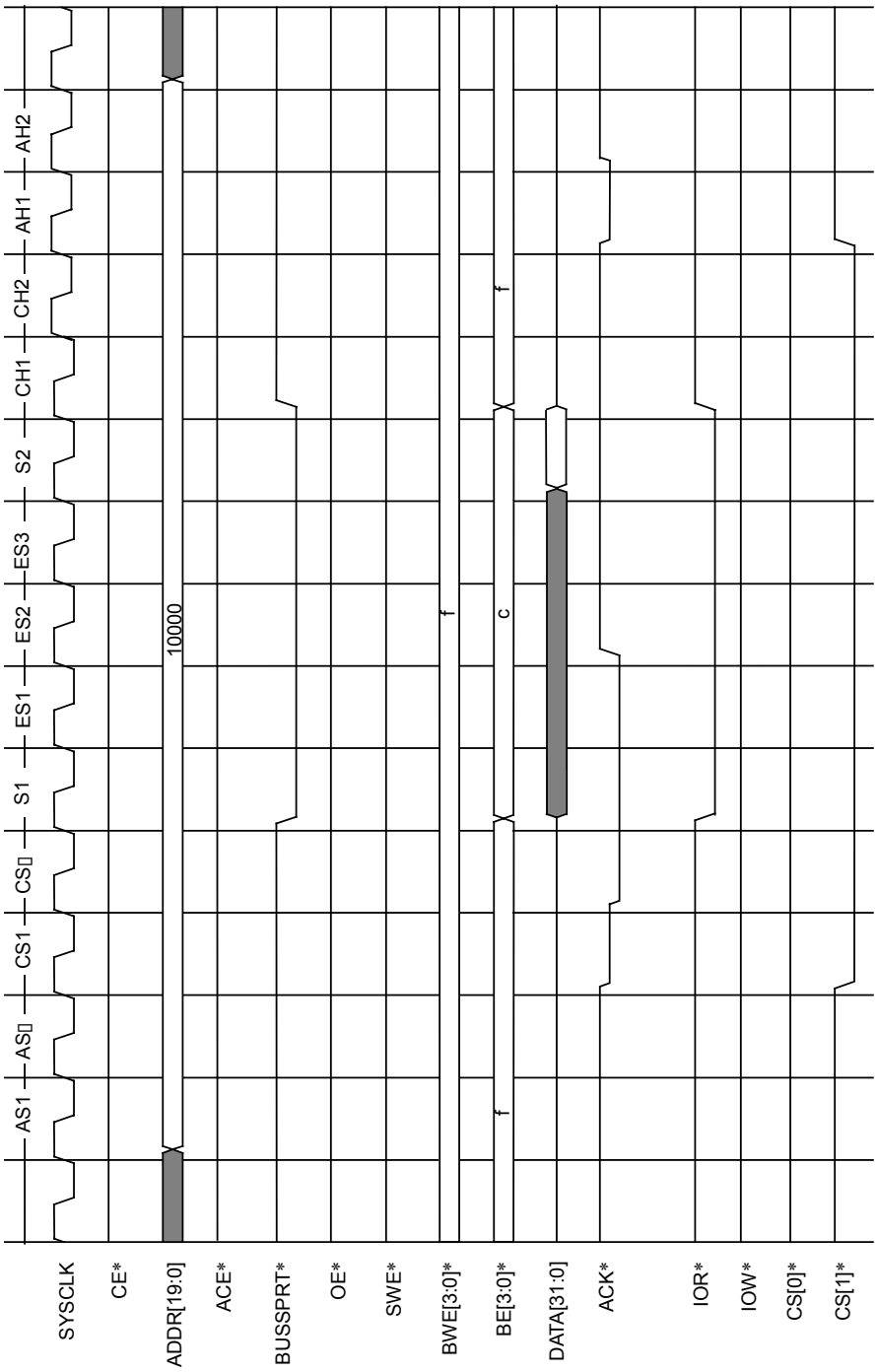


Figure 7.5.36 1-half-word Read Access for the ATA/PIO Space (CS1) (READY, SHWT=2, 16-bit Bus)

7.6 Flash ROM, SRAM Usage Example

Figure 7.6.1 illustrates example Flash ROM connections, and Figure 7.6.2 illustrates example SRAM connections. Also, Figure 7.6.3 illustrates example connections with the SDRAM and the bus separated.

Since connecting multiple memory devices such as SDRAM and ROM onto a single bus increases the load, 100 MHz class high-speed SDRAM access may not be performed normally. As a corrective measure, there is a way of reducing the bus load by connecting a device other than SDRAM via a buffer. If such a method is employed, directional control becomes necessary since the data becomes bidirectional.

The TX4938 prepares the BUSSPRT* signal for performing data directional control (see Figure 7.6.3). BUSSPRT* is asserted when the External Bus Controller channel is active and a Read operation is being performed.

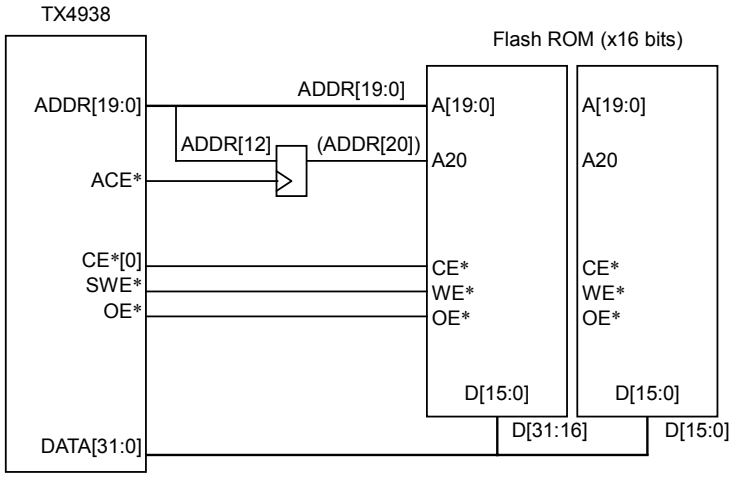


Figure 7.6.1 Flash ROM (x16 Bits) Connection Example (32-bit Data Bus)

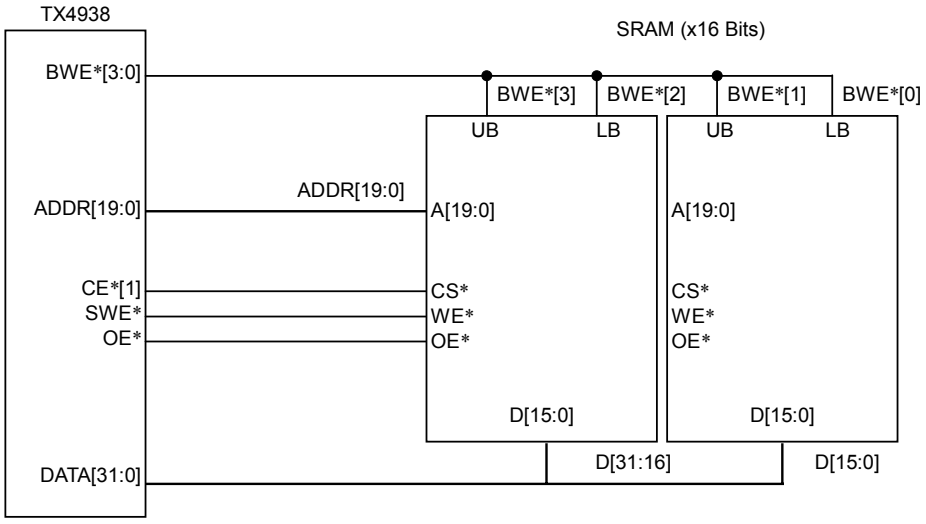


Figure 7.6.2 SRAM (x16 Bits) Connection Example (32-bit Data Bus)

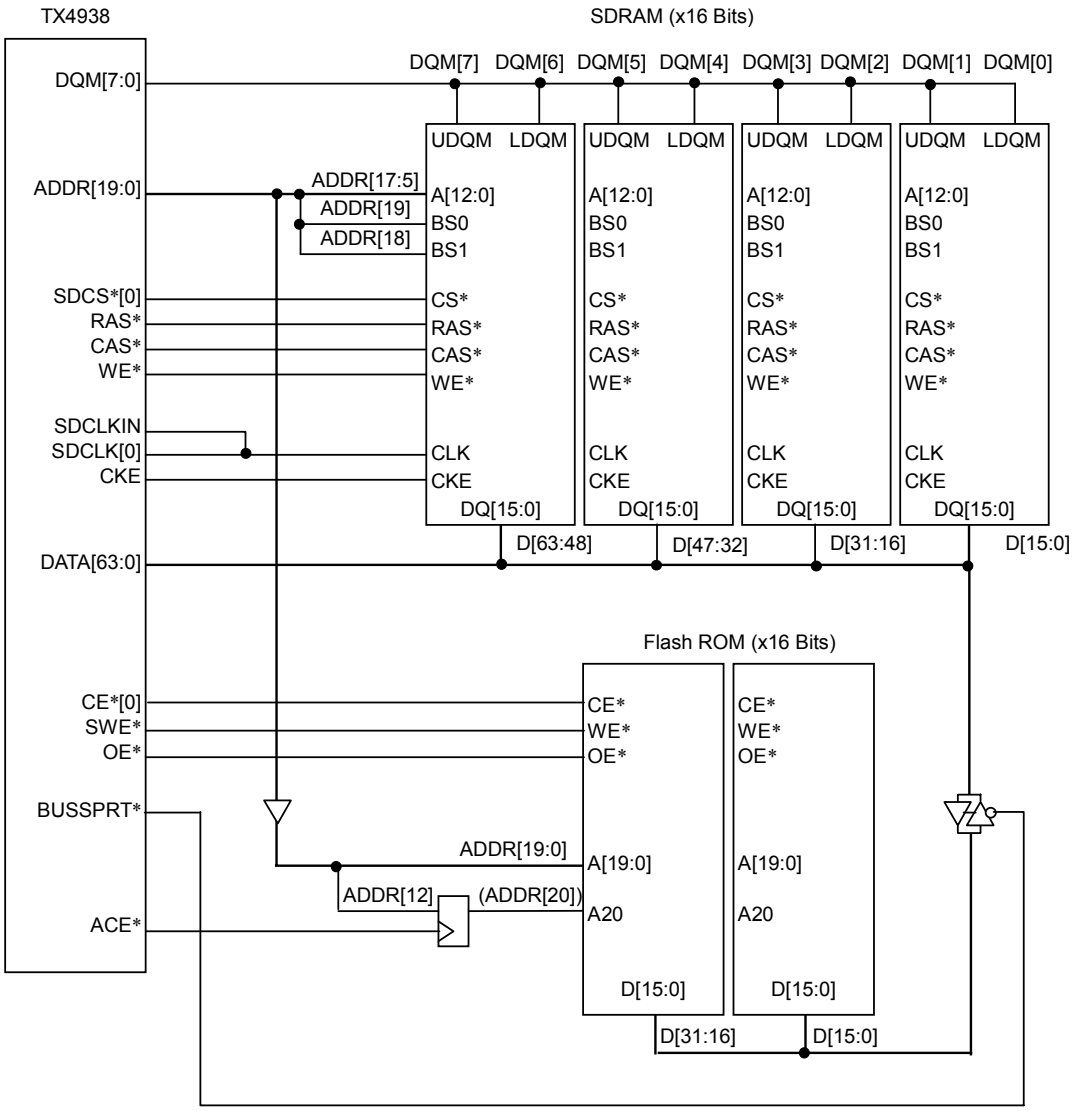


Figure 7.6.3 Connection Example with SDRAM and the Bus Separated

8. DMA Controller

8.1 Features

The TX4938 contains two four-channel DMA Controller (DMAC0, DMAC1) that executes DMA (Direct Memory Access) with memory and I/O devices.

The DMA Controller has the following characteristics.

<DMAC0, DMAC1>

- Has four on-chip DMA channels
- Supports Memory-Memory Copy modes that do not have address boundary limitations. Burst transfer of up to eight double words is possible for each Read or Write operation.
- Supports Memory Fill mode that writes double-word data to the specified memory region
- Supports Chained DMA Transfer
- On-chip signed 24-bit address count up registers for both the source address and destination address
- On-chip 26-bit Byte Count Register for each channel
- One of two methods can be selected for determining access priority among multiple channels: Round Robin or Fixed Priority
- Big Endian or Little Endian mode can be set separately for each channel

<DMAC0>

- Supports external I/O devices with 8-, 16-, and 32-bit Data Bus widths and transfer between memory devices.
- Supports single address transfer (Fly-by DMA) and dual address transfer when in the external I/O DMA Transfer Mode that is operated by external request signals
- Supports DMA on-chip Serial I/O Controllers

<DMAC1>

- Supports DMA on-chip AC-Link Controllers

8.2 Block Diagram

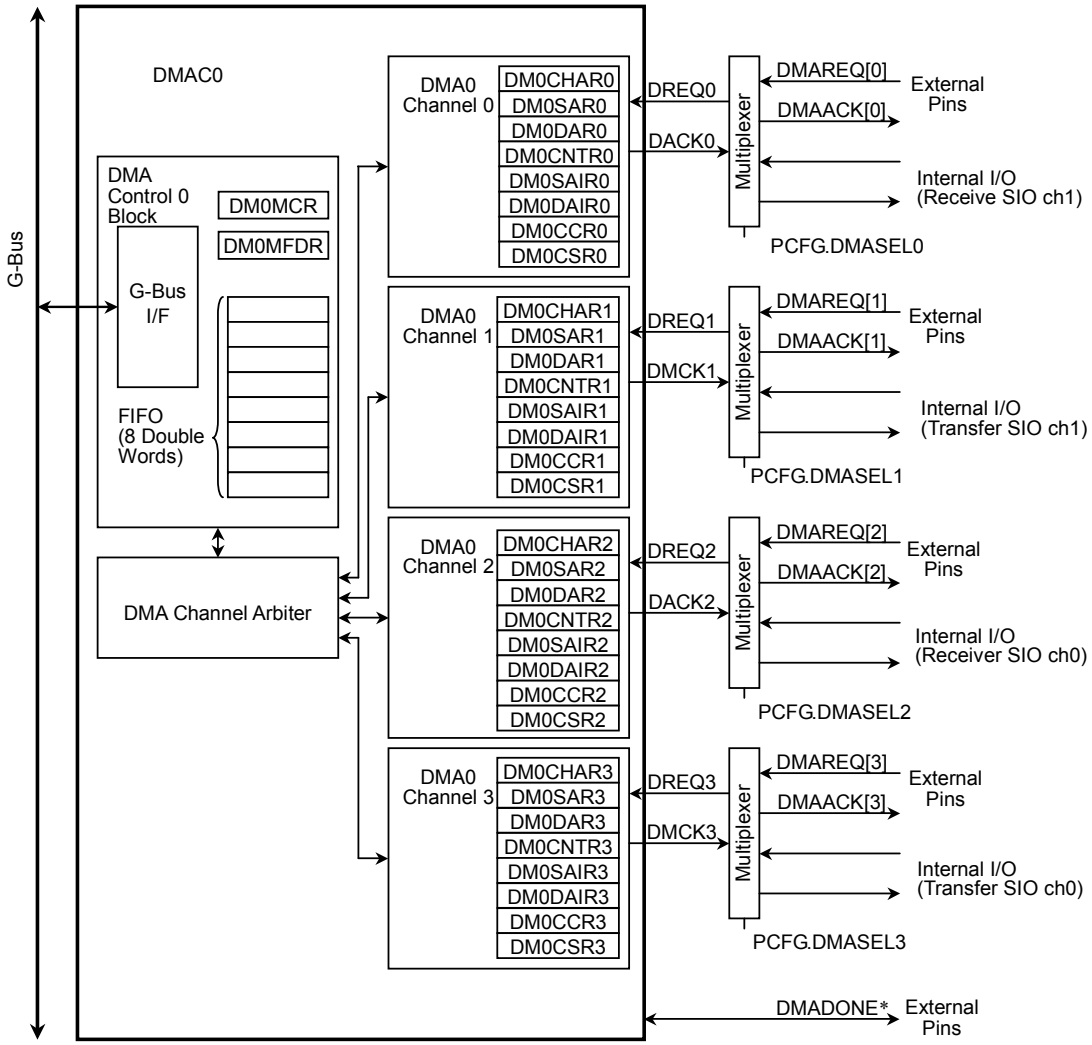


Figure 8.2.1 DMA0 Controller Block Diagram

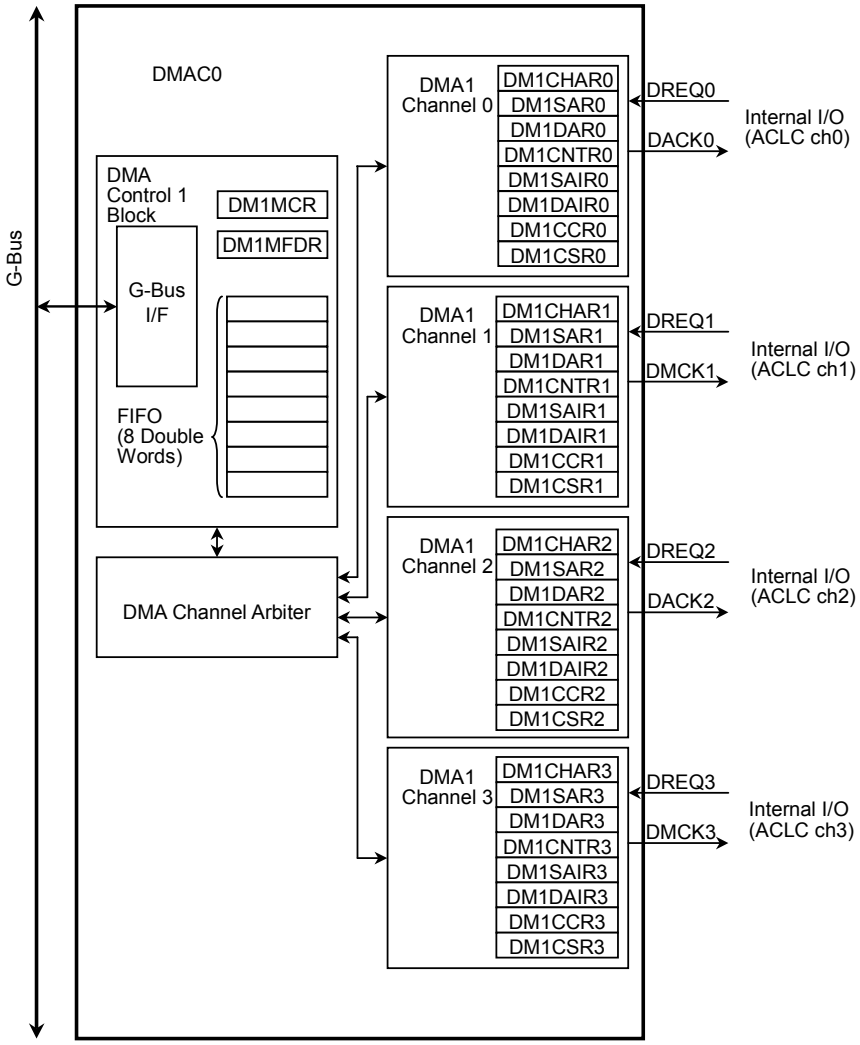


Figure 8.2.2 DMA1 Controller Block Diagram

8.3 Detailed Explanation

8.3.1 Transfer Mode

The DMA Controller (DMAC0, DMAC1) supports five transfer mode types (refer to Table 8.3.1 below). The setting of the External Request bit (DMCCRn.EXTRQ) of the DMA Channel Control Register selects whether transfer with an I/O device is a DMA transfer.

- I/O DMA Transfer Mode (DMCCRn.EXTRQ = "1")
  Perform DMA transfer with either an external device connected to the External Bus Controller or an on-chip I/O device (ACLC or SIO).

- Memory Transfer Mode (DMCCRn.EXTRQ = "0")
  Either copies data between memory devices or fills data in memory.

Table 8.3.1 DMA Controller Transfer Modes

DMA Controller	Transfer Mode	DMCCRn EXTREQ	PCFG DMASEL	DMCCRn SNGAD	DMSAR	DMDAR	Ref.
DMAC0	External I/O (Single Address)	1	0	1	√	—	8.3.3 8.3.7
	External I/O (Dual Address)	1	0	0	√	√	8.3.3 8.3.8
	Internal I/O (SIO)	1	1	0	√	√	8.3.4 8.3.8
	Memory-Memory Copy	0	—	0	√	√	8.3.4 8.3.8
	Fill Memory	0	—	1	√	—	8.3.6 8.3.7
DMAC1	Internal I/O (ACLC)	1	—	0	√	√	8.3.4 8.3.8
	Memory-Memory Copy	0	—	0	√	√	8.3.4 8.3.8
	Fill Memory	0	—	1	√	—	8.3.6 8.3.7

8.3.2 On-chip Registers

The DMA Controller has two shared registers that are shared by four channels. Section 8.4 explains each register in detail.

- Shared Registers
  DMMCR: DMA Master Control Register
  DMMFDR: DMA Memory Fill Data Register

- DMA Channel Register
  DMCHARn: DMA Chained Address Register
  DMSARn: DMA Source Address Register
  DMDARn: DMA Destination Address Register
  DMCNTRn: DMA Count Register
  DMSAIRn: DMA Source Address Increment Register
  DMDAIRn: DMA Destination Address Increment Register
  DMCCRn: DMA Channel Control Register
  DMCSRn: DMA Channel Status Register

8.3.3 External I/O DMA Transfer Mode

The External I/O DMA Transfer Mode performs DMA transfer with external I/O devices that are connected to the External Bus Controller.

8.3.3.1 External Interface

External I/O devices signal DMA requests to the DMA Controller by asserting the DMA Transfer Request Signal (DMAREQ[n]). On the other hand, the DMA Controller accesses external I/O devices by asserting the DMA Transfer Acknowledge Signal (DMAACK[n]).

The DMA Transfer Request signal (DMAREQ[n]) can use the Request Polarity bit (REQPOL) of the DMA Channel Control Register (DMCCRn) to select the signal polarity for each channel, and can use the Edge Request bit (EGREQ) to select either edge detection or level detection for each channel. The DMA Transfer Acknowledge signal (DMAACK[n]) can also use the Acknowledge Polarity bit (ACKPOL) to select the polarity.

Please assert/deassert the DMAREQ[n] signal as follows below.

- When level detection is set (DMCCRn.EGREQ = 0)

  The DMAREQ[n] signal must be continuously asserted until one SYSCLK cycle after the DMAACK[n] signal is asserted. Also, the DMAREQ[n] signal must be deasserted before the CE*/CS* signal is deasserted. If this signal is asserted too soon, DMA transfer will not be performed. If this signal is asserted or deasserted too late, unexpected DMA transfer may result.

  During Dual Address transfer, we recommend detecting assertion of the CE* signal for the external I/O device that is currently asserting DMAACK[n], then deasserting DMAREQ[n].

- When edge detection is set (DMCCRn.EGREQ = 1)

  Please set up assertion of the DMAREQ[n] signal so the DMAREQ[n] signal is asserted after the DMAACK[n] signal corresponding to a previously asserted DMAREQ[n] signal is deasserted. The DMAREQ[n] signal will not be detected even if it is asserted before DMAACK[n] is deasserted.

Figure 8.3.1 is a timing diagram that shows the timing of external DMA access. In this timing diagram, both the DMAREQ[n] signal and the DMAACK[n] signal are set to Low active (DMCCRn.REQPL = 0, DMCCRn.ACKPOL = 0).

The DMAACK[n] and DMADONE[n] signals, which are DMA control signals, are synchronized to SDCLK. When these signals are used by an external I/O device that is synchronous to SYSCLK, it is necessary to take clock skew into account.

The DMAACK[n] signal is asserted either at the SYSCLK cycle, the same as with assertion of the CE*/CS* signal, or before that. In addition, it is deasserted after the last ACK*/READY signal is deasserted.

When the DMADONE* signal (refer to 8.3.3.4) is used as an output signal, it is asserted for at least one SYSCLK cycle while the DMAACK[n] signal is asserted either during the same SYSCLK cycle that the CE*/CS* signal is deasserted or during a subsequent SYSCLK cycle. When the DMADONE* signal is used as an input signal, it must be asserted for one SYSCLK cycle while the DMAACK[n] signal is being asserted.

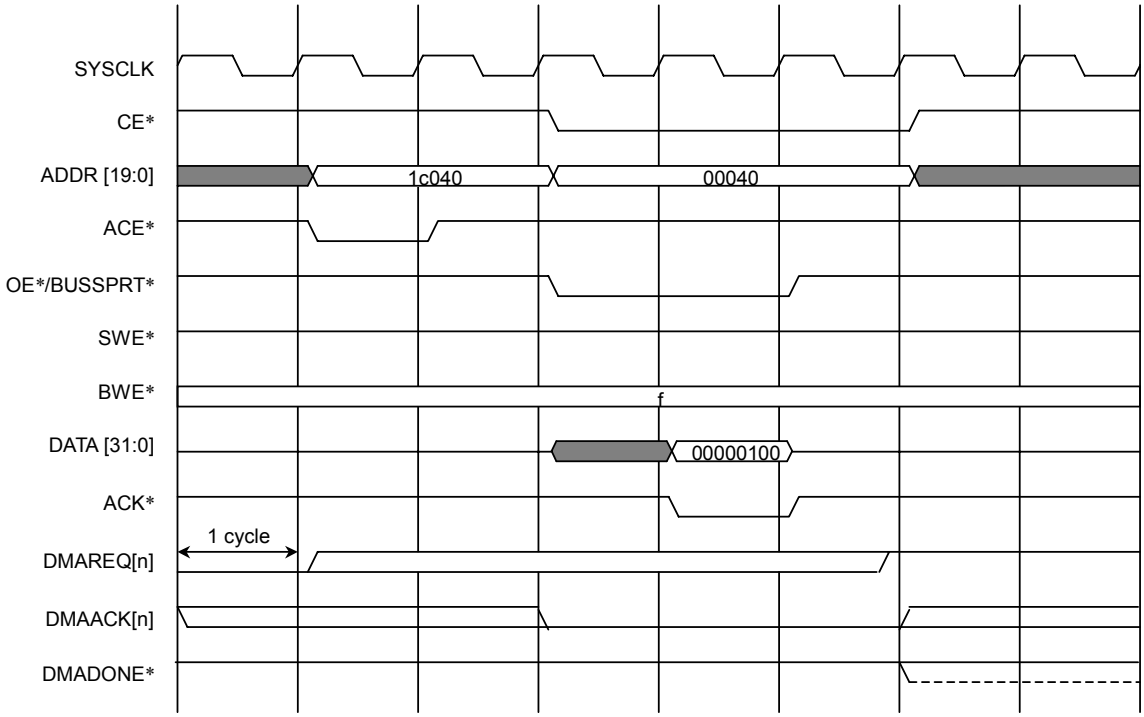


Figure 8.3.1 External I/O DMA Transfer (Single Address, Level Request)

8.3.3.2 Dual Address Transfer

If the Single Address bit (DMCCRn.SNGAD) has been cleared, access to external I/O devices and to external memory is each performed continuously. Each access is the same as normal access except when the DMAACK[n] signal is asserted.

Please refer to "8.3.8 Dual Address Transfer" for information regarding setting the register.

8.3.3.3 Single Address Transfer (Fly-by DMA)

If the Single Address bit (DMCCRn.SNGAD) is set, either data reading from an external I/O device and data writing to external memory or data reading from external memory and data writing to an external I/O device is performed simultaneously. The following conditions must be met in order to perform Single Address transfer.

- The data bus widths of the external I/O device and external memory match
- Data can be input/output to/from the external I/O device and external memory during the same clock cycle

The Transfer Direction bit (MEMIO) of the DMA Channel Control Register (DMCCRn) specifies the transfer direction.

- From memory to an external I/O device (DMCCRn.MEMIO = "1")

  External memory Read operation to an address specified by the DMA Source Address Register (DMSARn) is performed simultaneously to assertion of the DMAACK[n] signal.
- Single Address transfer from memory to an external I/O device (DMCCRn.MEMIO = "0")

  External memory Write operation to an address specified by the DMA Source Address Register (DMSARn) is performed simultaneously to assertion of the DMAACK[n] signal. At this time, the external I/O device drives the DATA signal instead of the TX4938.

Special attention must be paid to the timing design when the bus clock frequency is high or when performing Burst transfer. Single Address transfer using Burst transfer with SDRAM is not recommended.

8.3.3.4 DMADONE* Signal

The DMADONE* signal operates as either the DMA stop request input signal or the DMA done signalling output signal, or may operate as both of these signals depending on the setting of the DONE Control Field (DNCTRL) of the DMA Channel Control Register (DMCCRn).

The DMADONE* signal is shared by four channels. The DMADONE* channel is valid for a channel when the DMAACK[n] signal for that channel is asserted.

If the DMADONE* channel is set to be used as an output signal (DMCCRn.DNCTRL = 10/11), it will operate as follows depending on the setting of the Chain End bit (CHDN) of the DMA Channel Control Register (DMCCRn).

When the Chain End bit (CHDN) is set, the DMADONE* signal is only asserted when the DMAACK[n] signal for the last DMA transfer in the Link List Command Chain is asserted.

When the Chain End bit (CHDN) is cleared, the DMADONE* signal is asserted when the DMAACK[n] signal for the last data transfer in a DMA transfer specified by the current DMA Channel Register is asserted. Namely, if the Link List Command chain is used, there is one assertion at the end of each data transfer specified by each Descriptor.

If the DMADONE* signal is set to be used as an input signal (DMCCRn.DNCTRL = 01/11), DMA transfer can be set to end normally when the external device asserts the DMADONE* signal when the DMAACK[n] signal of channel *n* is asserted. DMADONE* is asserted during DMAACK[n] is not asserted, then unexpected operation occurs. When DMA transfer is terminated by the DMADONE* assertion of the external device, the External DONE Assert bit (DMCSRn.EXTDN) of the DMA Channel Status Register is set regardless of the setting of the Chain End bit (CHDN) of the DMA Channel Control Register (DMCCRn). Operation is as follows depending on the setting of the Chain End bit (CHDN).

When the Chain End bit (CHDN) is set, all DMA transfer for that chain is terminated. At this time, the Normal Chain End bit (NCHNC) and the Normal Transfer End bit (NTRNFC) of the DMA Channel Status Register are both set and the Transfer Active bit (DMCCRn.XFACT) of the DMA Channel Control Register is cleared.

When the Chain End bit (CHDN) is cleared, only DMA transfer specified by the current DMA Channel Register ends normally, and only the Normal Transfer End bit (NTRNFC) is set. When the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn) is set, chain transfer is executed and DMA transfer continues. When the Chain Enable bit (CHNEN) is cleared, the Transfer Active bit (DMCCRn.XFACT) is cleared and the Normal Chain End bit (NCHNC) is set.

Three clock cycles are required from external assertion of the DMADONE* signal to disabling of new DMA access. Operation will not stop even if the bus operation in progress is a Single transfer or a Burst transfer. For example, if the DMADONE* signal is asserted during Read operation of Dual Address transfer, the corresponding Write bus operation will also be executed.

If the DMADONE* pin is set to become both input and output for channel *n* (DMCCRn.DNCTRL = “11”), the DMADONE* signal becomes an open drain signal when the channel becomes active. When used by this mode, the DMADONE* signal must be pulled up by an external source. When in this mode, the External DONE Assert bit (DMCSRn.EXTDN) is not only set when asserted by an external device, but is also set when asserted by the TX4938.

8.3.4 Internal I/O DMA Transfer Mode

Performs DMA with the on-chip Serial I/O Controller and the AC-link Controller. Set the DMA Channel Control Register (DMCCRn) as follows.

- DMCCRn.EXTRQ = 1: I/O DMA Transfer mode
- DMCCRn.SNGAD = 0: Dual Address Transfer

Refer to “8.3.8 Dual Address Transfer” and “11.3.6 DMA transfer (Serial I/O Controller)” or “14.3.6.4 DMA operation (AC-link Controller) for more information.

8.3.5 Memory-Memory Copy Mode

It is possible to copy memory from any particular address to any other particular address when in the Memory-Memory Copy mode.

Set the DMA Channel Control Register (DMCCRn) as follows.

- DMCCRn.EXTRQ = 0: Memory Transfer mode
- DMCCRn.SNGAD = 0: Dual Address mode

Furthermore, when in the Memory-Memory Copy mode it is possible to set the interval for requesting ownership of each bus using the Internal Request Delay field (INTRQD) of the DMA Channel Control Register (DMCCRn).

Refer to "8.3.8 Dual Address Transfer" for information regarding the setting of other registers.

8.3.6 Memory Fill Transfer Mode

When in the Memory Fill Transfer mode, double word data set in the DMA Memory Fill Data Register (DMMFDR) is written to the data region specified by the DMA Source Address Register (DMSARn). This data can be used for initializing the memory, etc.

Set the DMA Channel Control Register (DMCCRn) as follows.

- DMCCRn.EXTRQ = 0: Memory transfer mode
- DMCCRn.SNGAD = 1: Single Address Transfer
- DMCCRn.MEMIO = 0: Transfer from I/O to memory

In addition, when in the Memory Fill Transfer mode, it is possible to set the interval for requesting ownership of each bus using the Internal Request Delay field (INTRQD) of the DMA Channel Control Register (DMCCRn).

Refer to "8.3.7 Single Address Transfer" for information regarding the setting of other registers.

By using this function together with the memory Write function that writes to multiple SDRAM Controller memory channels simultaneously (refer to Section 9.3.4), it is possible to initialize memory even more efficiently.

8.3.7 Single Address Transfer

This section explains register settings during Single Address transfer (DMCCRn.SNGAD = 1). This applies to the following DMA Transfer modes.

- External I/O (Single Address) Transfer
- Memory Fill Transfer

8.3.7.1 Channel Register Settings During Single Address Transfer

Table 8.3.2 shows restrictions of the Channel Register settings during Single Address transfer. If these restrictions are not met, then a Configuration Error is detected, the Configuration Error bit (CFERR) of the DMA Channel Status Register (DMCSRn) is set and DMA transfer is not performed.

For Burst transfer, +8, 0, or –8 can be set to the DMA Source Address Increment Register (DMSAIRn). Setting 0 is only possible during transfer from memory to external I/O. A Configuration Error will result if the value "0" is set during transfer from external I/O to memory or during Memory Fill transfer.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer setting size is 2 bytes or larger, then set the DMA Source Address Register (DMSARn) with 1 to 3 low-order bits complemented.

- If the transfer size is 2 bytes, set the DMSARn with the low-order 1 bit complemented.
- If the transfer size is 4 bytes, set the DMSARn with the low-order 2 bits complemented.
- If the transfer size is 8 bytes or larger, set the DMSARn with the low-order 3 bits complemented.

Example: When the transfer address is 0x0_0001_0000, the DMA Source Address Register (DMSARn) is as follows below.

- DMSAIRn setting is "0" or greater: 0x0_0001_0000
- DMSAIRn setting is a negative value: 0x0_0001_0007

During Single Address transfer, the DMA Destination Address Register (DMDARn) and DMA Destination Address Increment Register (DMDAIRn) settings are ignored.

Table 8.3.2 Channel Register Setting Restrictions During Single Address Transfer

Transfer Setting Size (DMCCRn.XFSZ)	DMSARn[2:0]		DMSAIRn[2:0]	DMCNTRn[2:0]
	DMSAIRn is "0" or greater	DMSAIRn setting is a negative value		
1 Byte	***	***	***	***
2 Bytes	**0	**0	**0	**0
4 Bytes	*00	*00	*00	*00
8 Bytes	000	111	000	000
4 Double Words	000	111	8/0/-8	000
8 Double Words				
16 Double Words				
32 Double Words				

8.3.7.2 Burst Transfer During Single Address Transfer

According to the SDRAM Controller and External Bus Controller specifications, the DMA Controller cannot perform Burst transfer that spans across 32-double word boundaries. Consequently, if the address that starts DMA transfer is not a multiple of the transfer setting size (DMCCRn.XFSZ) (is not aligned), transfer cannot be performed by any of the transfer sizes that were specified by a Burst transfer. Therefore, the DMA Controller executes multiple Burst transactions of a transfer size smaller than the specified transfer size. This division method changes according to the seting of the Transfer Size Mode bit (DMCCRn.USEXFSZ) of the DMA Channel Control Register.

Figure 8.3.2 shows the Single Address Burst transfer status when the lower 8 bits of the Transfer Start address are 0xA8 and the transfer setting size (DMCCRn.XFSZ) is set to 4 double words.

Panel (a) of this figure shows the situation when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is "0". In this case, first a three-double word transfer is performed up to the address aligned to the transfer setting size. Then, four-double word transfer specified by the transfer setting size is repeated. This setting is normally used.

On the other hand, panel (b) shows when the Transfer Size Mode bit (DMCCRn.USEXFSWZ) is "1". In this case, transfer is repeated according to the transfer setting size. Three-double word transfer and one-double word transfer is only performed consecutively without releasing bus ownership when transfer spans across a 32-double word boundary.

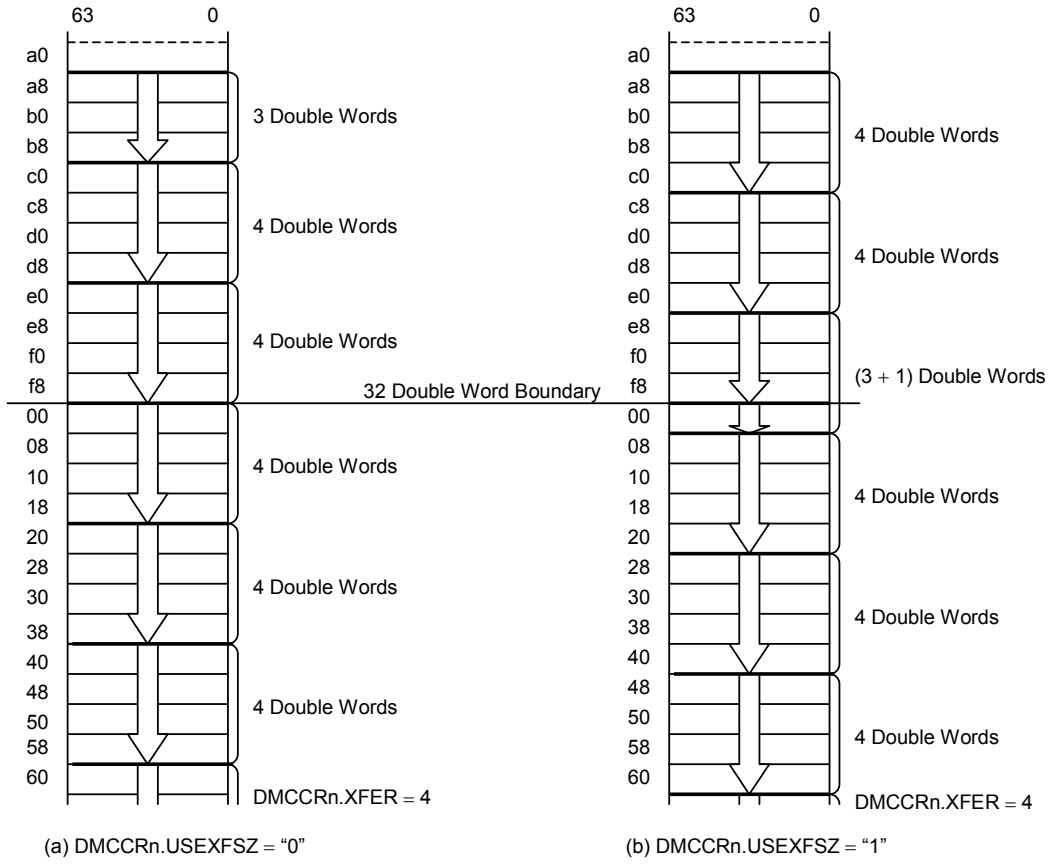


Figure 8.3.2 Non-aligned Single Address Burst Transfer

8.3.8 Dual Address Transfer

This section explains the register settings for Dual Address transfer (DMCCRn.SNGAD = 0). This applies to the following DMA transfer modes.

- External I/O (Dual Address) transfer
- Internal I/O DMA transfer
- Memory-Memory Copy transfer

8.3.8.1 Channel Register Settings During Dual Address Transfer

Table 8.3.3 shows restrictions of the Channel Register settings during Dual Address transfer. If these restrictions are not met, then a Configuration Error is detected, the Configuration Error bit (CFERR) of the DMA Channel Status Register (DMCSRn) is set, and DMA transfer is not performed.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer setting size is 8 bytes or larger, then a value will be set in the DMA Source Address Register (DMSARn) that reflects as follows.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer size is 2 bytes or larger, set the DMA Source Address Register (DMSARn) as follows:

- If the transfer size is 2 bytes, set the DMSARn with the low-order 1 bit complemented.
- If the transfer size is 4 bytes, set the DMSARn with the low-order 2 bits complemented.
- If the transfer size is 8 bytes or larger, set the DMSARn with the low-order 3 bits complemented.

Likewise, if the setting of the DMA Destination Address Increment Register (DMDAIRn) is negative and the transfer size is 2 bytes or larger, set the DMA Destination Address Register (DMDARn) as follows:

- If the transfer size is 2 bytes, set the DMDARn with the low-order 1 bit complemented.
- If the transfer size is 4 bytes, set the DMDARn with the low-order 2 bits complemented.
- If the transfer size is 8 bytes or larger, set the DMDARn with the low-order 3 bits complemented.

Example: When the transfer address is 0x0_0001_0000, the DMA Source Address Register (DMSARn) is as follows below.

- DMSAIRn setting is “0” or greater: 0x0_0001_0000
- DMSAIRn setting is a negative value: 0x0_0001_0007

Table 8.3.3 Channel Register Setting Restrictions During Dual Address Transfer

Transfer Setting Size (DMCCRn.XFSZ)	DMSARn[2:0]		DMDARn[2:0]		DMSAIRn	DMDAIRn	DMCNTRn	DMCCRn REVBYTE
	DMSAIRn setting is 0 or greater	DMSAIRn setting is a negative value	DMDAIRn setting is 0 or greater	DMDAIRn setting is a negative value				
1 Byte	***	***	***	***	***	***	***	0
2 Bytes	**0	**0	**0	**1	**0	**0	**0	0
4 Bytes	*00	*00	*00	*11	*00	*00	*00	0
8 Bytes, 4 / 8 Double Wods (DMMCR.FIFUM[n]=0)	000	111	000	111	000	000	000	0/1
4 / 8 Double Words (DMMCR.FIFUM[n]=1)	000	111	000	111	8/0/-8 †	8/-8 ‡	000	0/1
	***	—	***	—	8	8	***	0
	—	***	—	***	-8	-8		0
16 Double Words	Cannot be set (Configuration Error)							
32 Double Words	Cannot be set (Configuration Error)							

†: When DMSAIRn is set to 0, read access from source device is performed only one time per transmission specified by DMCCRn.XFSZ. For this reason, transfer can not be performed burst transfer to the I/O device which performs FIFO operation.

‡: 8, 0, or -8 can be specified when the Destination Burst Inhibit bit (DMCCRn.DBINH) is set.

8.3.8.2 Burst Transfer During Dual Address Transfer

The DMA Controller has a 64-bit 8-stage FIFO on-chip that is connected to the internal bus (G-Bus) for Burst transfer during Dual Address transfer. Since this FIFO employs a shifter, it is possible to perform transfer of any address or data size. Burst transfer is only performed when 4 Double Words or 8 Double Words is set by the Transfer Setting Size field (DMCCRn.XFSZ) and the FIFO Use Enable bit (DMMCRn.FIFUM[n]) of the DMA Master Control Register is set.

According to the SDRAM Controller and External Bus Controller specifications, the DMA Controller cannot perform Burst transfer that spans across 32-double word boundaries. Consequently, if the address that starts DMA transfer is not a multiple of the transfer setting size (DMCCRn.XFSZ) (is not aligned), transfer cannot be performed by any of the transfer sizes that were specified by a Burst transfer. Therefore, it is necessary to divide the transfer into multiple Burst transactions of a transfer size smaller than the specified transfer size. This division method changes according to the seting of the Transfer Size Mode bit (DMCCRn.USEXFSZ) of the DMA Channel Control Register and whether or not the address offset relative to the Transfer Setting size (DMCCRn.XFSZ) is equivalent to the source address and destination address combined.

Figure 8.3.3 shows Dual Address Burst transfer when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is set to “1”, the lower 8 bits of the Transfer Start address for the transfer source are set to 0xA8, the lower 8 bits of the Transfer Start address for the transfer destination are set to 0x38, and the Transfer Setting Size (DMCCRn.XFSZ) is set to 8 Double Words.

Transfer repeats according to the transfer setting size, regardless of the different address offsets. However, transfers that span across 32-double word boundaries are divided. Since data remains in the on-chip FIFO when in this mode, it becomes possible to share the on-chip FIFO among multiple DMA channels.

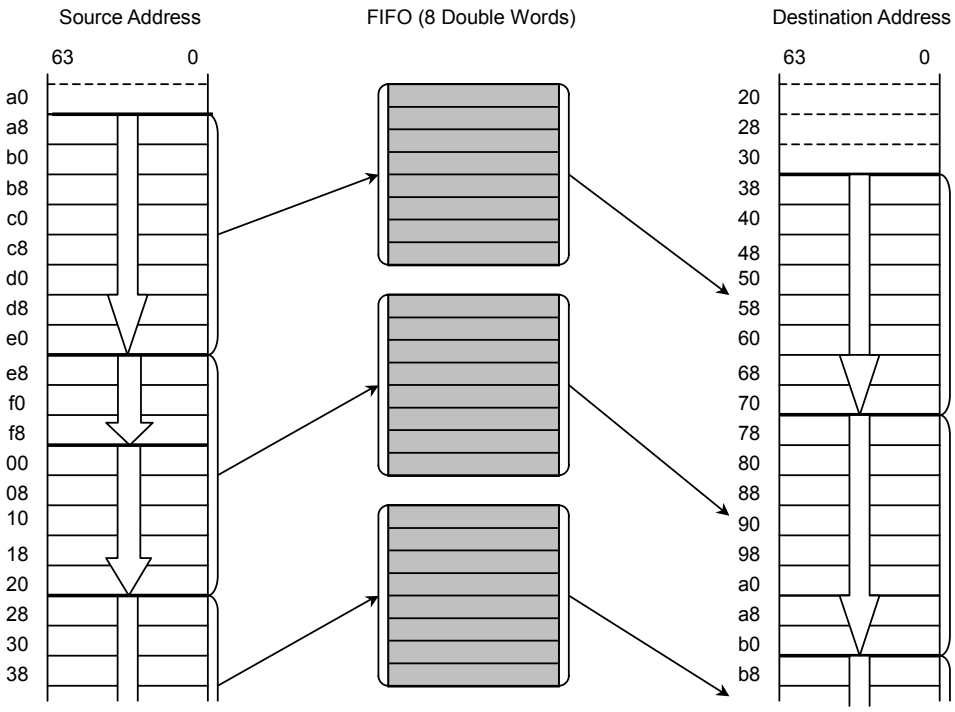


Figure 8.3.3 Dual Address Burst Transfer (DMCCRn.USEXFSZ = 1)

Figure 8.3.4 shows Dual Address Burst transfer when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is set to "0", the lower 8 bits of the Transfer Start address for the transfer source are set to 0xA8, the lower 8 bits of the Transfer Start address for the transfer destination are set to (a) 0x28/(b) 0x30, and the Transfer Setting Size (DMCCRn.XFSZ) is set to 8 double words.

Panel (a) of this figure shows when the address offset is equivalent. In this case, first transfer of three double words is performed up to the address that is aligned with the transfer setting size. Then, transfer of eight double words that is specified by the transfer setting size is repeated.

On the other hand, panel (b) show when the address offset is not equivalent. In this case, first only data up to the address that is aligned with the transfer setting size is read to the on-chip FIFO. Then, data is written up to the address that is aligned with the transfer setting size as long as data remains in the on-chip FIFO. Efficiency decreases since the transfer size is divided. Also, since data may remain in the on-chip FIFO, Burst transfer of a Dual Address that uses the on-chip FIFO simultaneously with another channel cannot be performed.

Using the Burst Inhibit bit makes it possible to mix Burst transfer with 8-Double-Word Single transfer. This in turn makes it possible to perform Burst access only for memory access during DMA transfer with external I/O devices that cannot perform Burst transfer.

When the Source Burst Inhibit bit (DMCCRn.SBINH) is set, data read from the Source Address to the on-chip FIFO is divided into multiple 8-byte Single Read transfers, then transfer is executed.

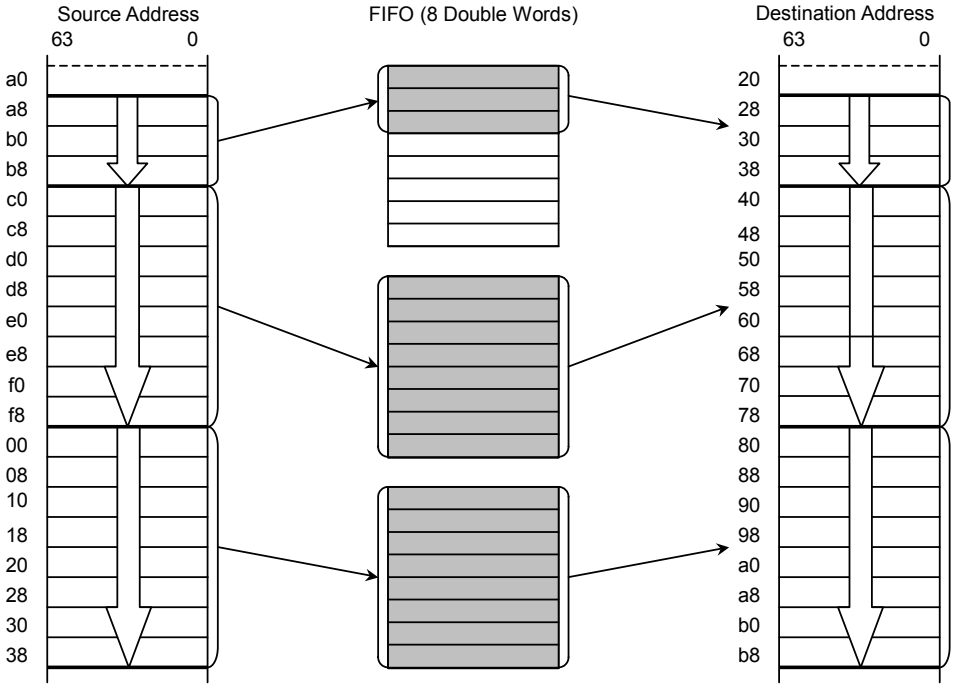
When the Destination Burst Inhibit bit (DMCCRn.DBINH) is set, data written from the FIFO to the Destination Address is divided into multiple 8-byte Single Write transfers, then transfer is executed.

When the Burst Inhibit bit is set, the TX4938 always performs an 8-byte Single transfer. For accesses to an external I/O device, a Single transfer is divided into multiple accesses, depending on its bus width. Thus, the address changes during a Single transfer. For more on this, see Section 7.3.5, "Data Bus Size." To continually access a fixed address in an external I/O device, program the trasnfer size (DMCCRn.XFSZ) to the bus width of the I/O device and perform Single transfers with the Burst Inhibit bit cleared.

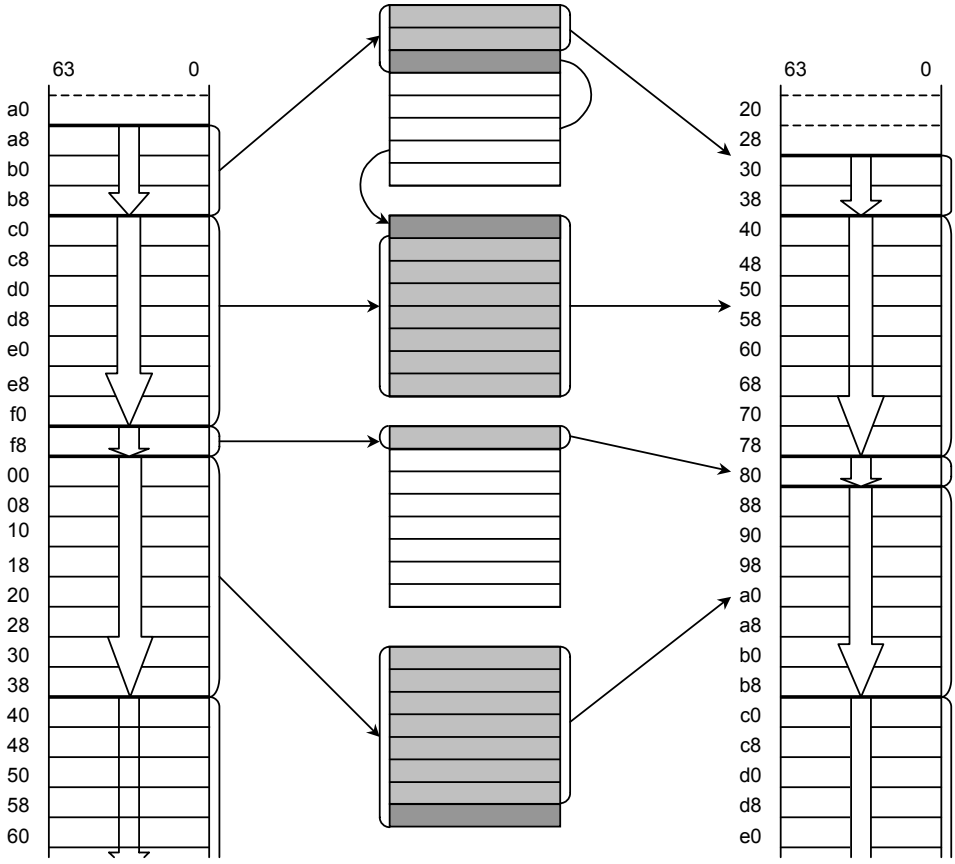
8.3.8.3 Double Word Byte Swapping

When the Reverse Byte bit (REVBYTE) of the DMA Channel Configuration Register (DMCCRn) is set, read double word data is written after byte swapping is performed. For example, if the read data is “0x01234567_90ABCDEF”, then the data “0xEFCDAB89_67452301” is written.

The Reverse Byte bit can only be set when the REVBYTE column of Table 8.3.3 is set so “0/1” is indicated.



(a) Address offset is equivalent



(b) Address offset differs

Figure 8.3.4 Dual Address Burst Transfer (DMCCRn.USEXFSZ = 0)

8.3.9 DMA Transfer

The sequence of DMA transfer that uses only the DMA Channel Register is as follows below.

1. Select DMA request signal
   When performing external I/O or internal I/O DMA, set the DMA Request Select field (PCFG.DMASEL) of the Pin Configuration Register.

2. Set the Master Enable bit
   Set the Master Enable bit (DMMCR.MSTEN) of the DMA Master Control Register.

3. Set the Address Register and Count Register
   Set the five following register values.
   - DMA Source Address Register (DMSARn)
   - DMA Destination Address Register (DMDARn)
   - DMA Count Register (DMCNTRn)
   - DMA Source Address Increment Register (DMSAIRn)
   - DMA Destination Address Increment Register (DMDAIRn)

4. Set Chain Address Register
   Set "0" to the DMA Chain Address Register (DMCHARn).

5. Clear the DMA Channel Status Register (DMCSRn)
   Clear when status from the previous DMA transfer remains.

6. Set the DMA Channel Control Register (DMCCRn)

7. Initiate DMA transfer
   DMA transfer is started by setting the Transfer Active bit (XFACT) of the DMA Channel Control Register.

8. Signal completion
   When DMA data transfer ends normally, set the Normal Transfer Complete bit (NTRNFC) of the DMA Channel Status Register (DMCSRn). An interrupt is signalled if the Transfer Complete Interrupt Enable bit (INTENT) of the DMA Channel Control Register (DMCCRn) is set.

   If an error is detected during DMA transfer, the error cause is recorded in the lower four bits of the DMA Channel Status Register and the transfer is interrupted. If the Error Interrupt Enable bit (INTENE) of the DMA Channel Control Register is set, then the interrupt is signaled.

8.3.10 Chain DMA Transfer

Table 8.3.4 shows the data structure in memory that the DMA Command Descriptor has. When the Simple Chain bit (SMPCHN) of the DMA Channel Control Register (DMCCRn) is set, only the initial four double words are used. DMSAIRn, DMDAIR, DMCCRn, and DMCSRn use the settings from when DMA started. In addition, all eight double words are used when the Simple Chain bit (SMPCHN) is cleared.

Saving the start memory address of another DMA Command Descriptor in the Offset 0 Chain Address field makes it possible to construct a chain list of DMA Command Descriptors (Figure 8.3.5). Set "0" in the Chain Address field of the DMA Command Descriptor at the end of the chain list.

When DMA transfer that is specified by one DMA Command Descriptor ends, the DMA Controller automatically reads the next DMA Command Descriptor indicated by the Chain Address Register (Chain transfer), then continues DMA transfer. Continuous DMA transfer that uses multiple Descriptors connected into such a chain-like structure is called Chain DMA transfer.

Since the DMA Channel Status Register is also overwritten during Chain transfer when the DMA Simple Chain bit (SMPCHN) is cleared, be sure not to unnecessarily clear necessary bits.

Placing DMA Command Descriptors at addresses that do not span across 32-double-word boundaries in memory is efficient since they are read by one G-Bus Burst Read operation.

Table 8.3.4 DMA Command Descriptors

Offset Address	Field Name	Transfer Destination Register
0x00	Chain Address	DMA Chain Address Register (DMCHARn)
0x08	Source Address	DMA Source Address Register (DMSARn)
0x10	Destination Address	DMA Destination Address Register (DMDARn)
0x18	Count	DMA Count Register (DMCNTRn)
0x20	Source Address Increment	DMA Source Address Increment Register (DMSAIRn)
0x28	Destination Address Increment	DMA Destination Address Increment Register (DMDAIRn)
0x30	Channel Control	DMA Channel Control Register (DMCCRn)
0x38	Channel Status	DMA Channel Status Register (DMCSRn)

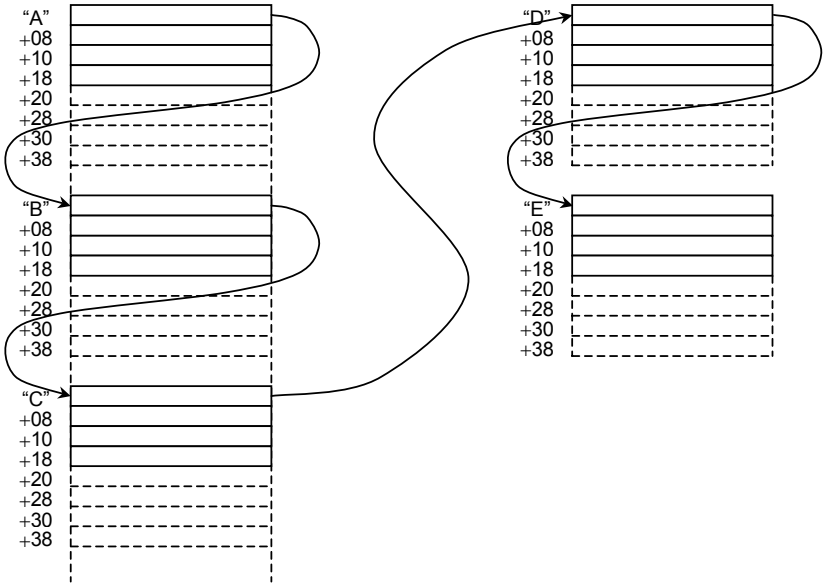


Figure 8.3.5 DMA Command Descriptor Chain

The sequence of Chain DMA transfer is as follows below.

1. Select DMA request signal
   When performing external I/O or internal I/O DMA, set the DMA Request Select field (PCFG.DMASEL) of the Pin Configuration Register.

2. Set the Master Enable bit
   Set the Master Enable bit (DMMCR.MSTEN) of the DMA Master Control Register.

3. Structure of the DMA command Descriptor chain
   Construct the DMA Command Descriptor Chain in memory.

4. Set the Count Register
   Set "0" to the DMA Count Register (DMCNTRn) .
   Sets the DMA Source Address Increment Register (DMSAIRn) and DMA destination Address Increment Register (MMDAIRn).

5. Clear the DMA Channel Status Register (DMCSRn)
   Clear the status of the previous DMA transfer.

6. Set the DMA Channel Control Register (DMCCRn).

7. Initiate DMA transfer
   Setting the address of the DMA Command Descriptor at the beginning of the chain list in the DMA Chain Address Register (DMCHARn) automatically initiates DMA transfer. First, the value stored in each field of the DMA Command descriptor at the beginning of the Chain List is read to each corresponding DMA Channel register (Chain transfer), then DMA transfer is performed according to the read value.

   When a value other than "0" is stored in the DMA Chain Address Register (DMCHARn), data of the size stored in the DMA Count Register (DMCNTRn) is completely transferred, then the DMA Command Descriptor value of the memory address specified by the DMA Chain Address Register is read.

   In addition, if the Chain Address field value read the Descriptor 0, the DMA Chain Address Register value is not updated. All previous values (Data Command Descriptor Addresses with the value "0" in the Chain Address field when the values were read) are held.

   0 Value judgement is performed when the lower 32 bits of the DMA Chain Address Register are rewritten. If the value is not "0" at this time, DMA transfer is automatically initiated. Therefore, please write to the upper 32 bits first when writing to the DMA Chain Address Register using 32-bit Store instructions.

8. Signal completion
   Set the Normal Chain End bit (NCHNC) of the DMA Channel Status Register (DMCSRn) when DMA data transfer of all Descriptor Chains is complete. An interrupt is signalled if the Chain End Interrupt Enable bit (INTENC) of the DMA Channel Control Register (DMCCRn) is set at this time.

   In addition, the Normal Transfer End bit (NTRNFC) of the DMA Channel Status Register (DMCSRn) is set each time DMA data transfer specified by each DMA Command Descriptor ends normally. An interrupt is signalled if the Transfer End Interrupt Enable bit (INTENT) of the DMA Channel Control Register (DMCCRn) is set at this time.

   If an error is detected during DMA transfer, the error cause is recorded in the lower four bits of the DMA Channel Status register and transfer is interrupted. An interrupt is signalled if the Error Interrupt Enable bit (INTENE) of the DMA Channel Control Register is set.

8.3.11 Dynamic Chain Operation

It is possible to add DMA Command Descriptor chains to the DMA Command Descriptor chain while Chain DMA transfer is in progress. This is performed according to the following procedure.

1. Construct the DMA Command Descriptor chain
   Construct the DMA Command Descriptor chain to be added to memory.

2. Add a DMA Command Descriptor chain
   Substitute the address of the Command Descriptor at the beginning of the Descriptor Chain to be added into the Chain Address field of the Descriptor at the end of the DMA Command Descriptor chain that is currently performing DMA transfer.

3. Check the Chain Enable bit
   Read the value of the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn). If that value is "0", then write the Chain Address field value of the DMA Command Descriptor that is indicated by the address stored in the DMA Chain Address Register (DMCHARn).

8.3.12 Interrupts

An interrupt number (10 – 13) of the Interrupt Controller is mapped to each channel. In addition, there are completion interrupts for when transfer ends normally and error interrupts for when transfer ends abnormally for each channel. When an interrupt occurs, then the bit that corresponds to either the Normal Interrupt Status field (DIS[3:0]) or the Error Interrupt Status field (EIS[3:0]) of the DMA Master Control Register (DMMCR) is set.

Figure 8.3.6 shows the relationship between the Status bit and Interrupt Enable bit for each interrupt cause. Refer to the explanation for each Status bit for more information regarding each information cause.

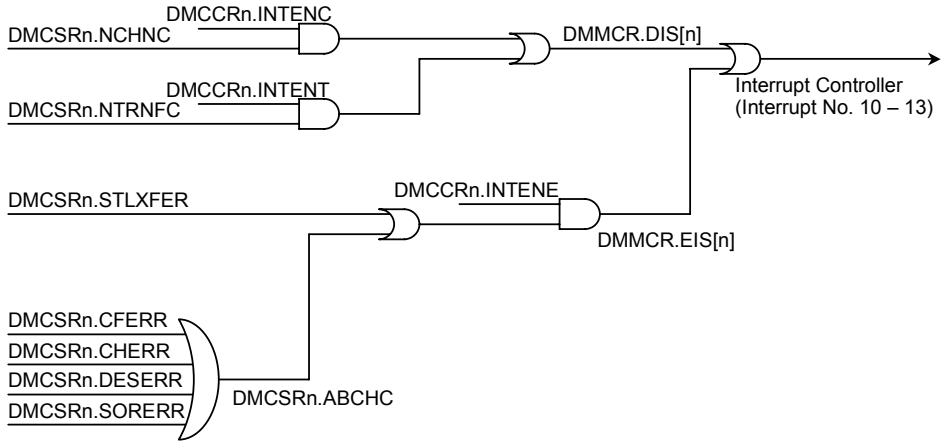


Figure 8.3.6 DMA Controller Interrupt Signal

8.3.13 Transfer Stall Detection Function

If the period from when a certain channel last performs internal bus access to when the next internal bus access is performed exceeds the Transfer Stall Detection Interval field (STLTIME) of the DMA Channel Control Register (DMCCRn), the Transfer Stall Detection bit (STLXFER) of the DMA Channel Status Register (DMCSRn) is set. An error interrupt is signalled if the Error Interrupt Enable bit (DMCCRn.INTENE) is set.

In contrast to other error interrupts, DMA transfer is not stopped. Normal DMA transfer is executed if bus ownership can be obtained. Furthermore, clearing the Transfer Stall Detection field (STLXFER) resumes transfer stall detection as well.

Setting the Transfer Stall Detection Interval field (STLTIME) to “000” disables the Transfer Stall Detection function.

8.3.14 Arbitration Among DMA Channels

The DMA Controller has an on-chip DMA Channel Arbiter that arbitrates bus ownership among four DMA channels that use the internal bus (G-Bus). There are two methods for determining priority: the round robin method and the fixed priority method. (See Figure 8.3.7.) The Round Robin Priority bit (RRPT) of the DMA Master Control Register (DMMCR) selects the priority method.

- Fixed priority (DMMCR.RRPT = 0)
  As shown below, Channel 0 has the highest priority and Channel 3 has the lowest priority.

  CH0 > CH1 > CH2 > CH3

- Round Robin method (DMMCR.RRPT = 1)
  The last channel to perform DMA transfer has the lowest priority.
  - After CH0 DMA transfer execution: CH1 > CH2 > CH3 > CH0
  - After CH1 DMA transfer execution: CH2 > CH3 > CH0 > CH1
  - After CH2 DMA transfer execution: CH3 > CH0 > CH1 > CH2
  - After CH3 DMA transfer execution: CH0 > CH1 > CH2 > CH3

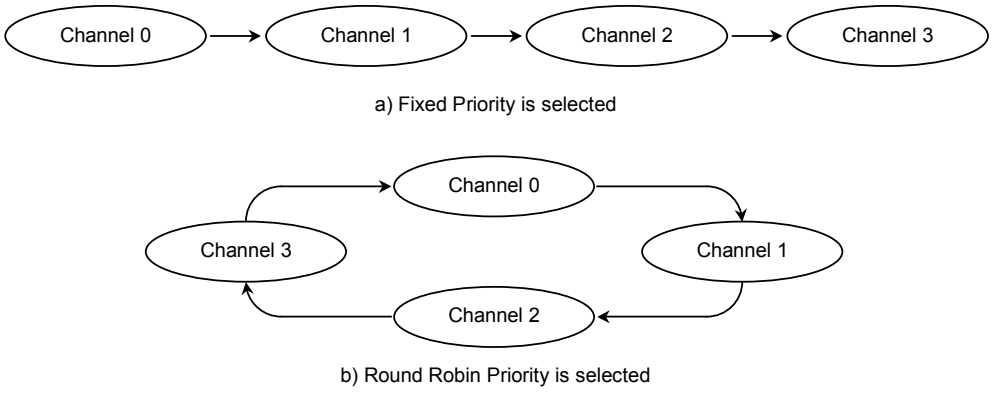


Figure 8.3.7 DMA Channel Arbitration

8.3.15 Restrictions in Access to PCI Bus

The PCI Controller detects a bus error if the DMA Controller performs one of the following accesses to the PCI Bus.

- Burst transfer exceeding 8 double words (PCICSTATUS.TLB)
- Address Increment value –8 Burst transfer (PCICSTATUS.NIB)
- Address Increment Value 0 Burst transfer (PCICSTATUS.ZIB)
- Dual Address Burst transfer when the setting for DMSARn, DMDARn, or DMCNTRn is not a double word boundary (PCICSTATUS.IAA)

In addition, Single Address transfers between an external I/O device and the PCI Bus are not supported. Data transfer is not performed, but no error is detected.

8.4 DMA Controller Registers

Table 8.4.1 DMA Controller 0 Registers

Offset Address	Bit Width	Mnemonic	Register Name
0xB000	64	DM0CHAR0	DMA Chain Address Register 0
0xB008	64	DM0SAR0	DMA Source Address Register 0
0xB010	64	DM0DAR0	DMA Destination Address Register 0
0xB018	64	DM0CNTR0	DMA Count Register 0
0xB020	64	DM0SAIR0	DMA Source Address Increment Register 0
0xB028	64	DM0DAIR0	DMA Destination Address Increment Register 0
0xB030	64	DM0CCR0	DMA Channel Control Register 0
0xB038	64	DM0CSR0	DMA Channel Status Register 0
0xB040	64	DM0CHAR1	DMA Chain Address Register 1
0xB048	64	DM0SAR1	DMA Source Address Register 1
0xB050	64	DM0DAR1	DMA Destination Address Register 1
0xB058	64	DM0CNTR1	DMA Count Register 1
0xB060	64	DM0SAIR1	DMA Source Address Increment Register 1
0xB068	64	DM0DAIR1	DMA Destination Address Increment Register 1
0xB070	64	DM0CCR1	DMA Channel Control Register 1
0xB078	64	DM0CSR1	DMA Channel Status Register 1
0xB080	64	DM0CHAR2	DMA Chain Address Register 2
0xB088	64	DM0SAR2	DMA Source Address Register 2
0xB090	64	DM0DAR2	DMA Destination Address Register 2
0xB098	64	DM0CNTR2	DMA Count Register 2
0xB0A0	64	DM0SAIR2	DMA Source Address Increment Register 2
0xB0A8	64	DM0DAIR2	DMA Destination Address Increment Register 2
0xB0B0	64	DM0CCR2	DMA Channel Control Register 2
0xB0B8	64	DM0CSR2	DMA Channel Status Register 2
0xB0C0	64	DM0CHAR3	DMA Chain Address Register 3
0xB0C8	64	DM0SAR3	DMA Source Address Register 3
0xB0D0	64	DM0DAR3	DMA Destination Address Register 3
0xB0D8	64	DM0CNTR3	DMA Count Register 3
0xB0E0	64	DM0SAIR3	DMA Source Address Increment Register 3
0xB0E8	64	DM0DAIR3	DMA Destination Address Increment Register 3
0xB0F0	64	DM0CCR3	DMA Channel Control Register 3
0xB0F8	64	DM0CSR3	DMA Channel Status Register 3
0xB148	64	DM0MFDR	DMA Memory Fill Data Register
0xB150	64	DM0MCR	DMA Master Control Register

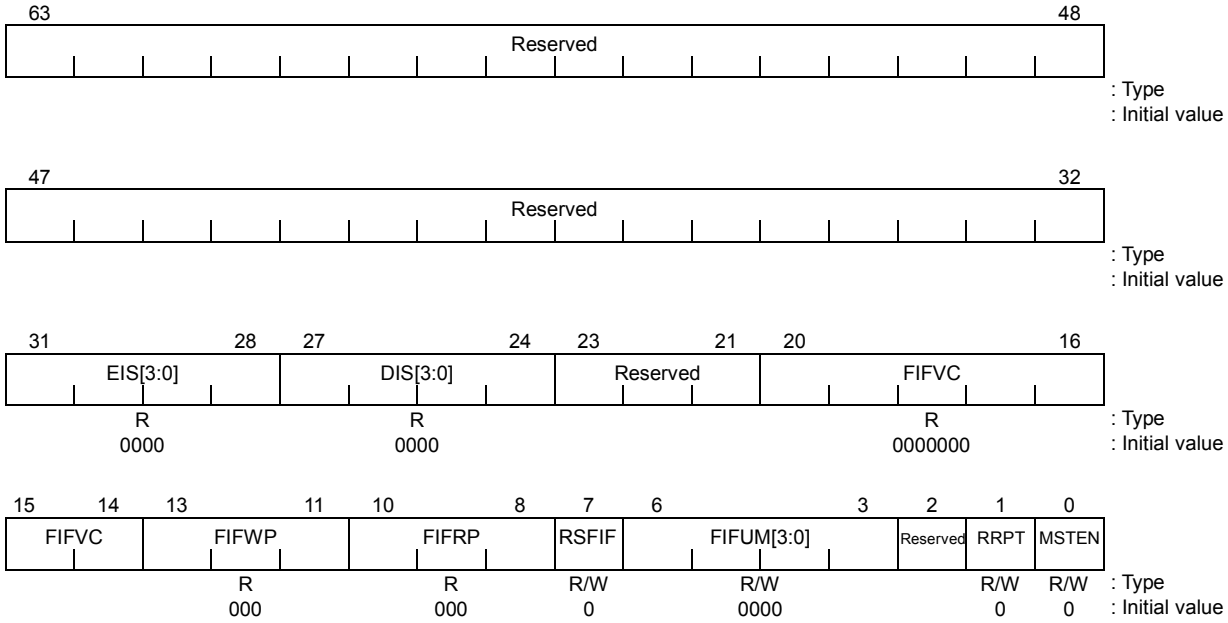
Table 8.4.2 DMA Controller 1 Registers

Offset Address	Bit Width	Mnemonic	Register Name
0xB800	64	DM1CHAR0	DMA Chain Address Register 0
0xB808	64	DM1SAR0	DMA Source Address Register 0
0xB810	64	DM1DAR0	DMA Destination Address Register 0
0xB818	64	DM1CNTR0	DMA Count Register 0
0xB820	64	DM1SAIR0	DMA Source Address Increment Register 0
0xB828	64	DM1DAIR0	DMA Destination Address Increment Register 0
0xB830	64	DM1CCR0	DMA Channel Control Register 0
0xB838	64	DM1CSR0	DMA Channel Status Register 0
0xB840	64	DM1CHAR1	DMA Chain Address Register 1
0xB848	64	DM1SAR1	DMA Source Address Register 1
0xB850	64	DM1DAR1	DMA Destination Address Register 1
0xB858	64	DM1CNTR1	DMA Count Register 1
0xB860	64	DM1SAIR1	DMA Source Address Increment Register 1
0xB868	64	DM1DAIR1	DMA Destination Address Increment Register 1
0xB870	64	DM1CCR1	DMA Channel Control Register 1
0xB878	64	DM1CSR1	DMA Channel Status Register 1
0xB880	64	DM1CHAR2	DMA Chain Address Register 2
0xB888	64	DM1SAR2	DMA Source Address Register 2
0xB890	64	DM1DAR2	DMA Destination Address Register 2
0xB898	64	DM1CNTR2	DMA Count Register 2
0xB8A0	64	DM1SAIR2	DMA Source Address Increment Register 2
0xB8A8	64	DM1DAIR2	DMA Destination Address Increment Register 2
0xB8B0	64	DM1CCR2	DMA Channel Control Register 2
0xB8B8	64	DM1CSR2	DMA Channel Status Register 2
0xB8C0	64	DM1CHAR3	DMA Chain Address Register 3
0xB8C8	64	DM1SAR3	DMA Source Address Register 3
0xB8D0	64	DM1DAR3	DMA Destination Address Register 3
0xB8D8	64	DM1CNTR3	DMA Count Register 3
0xB8E0	64	DM1SAIR3	DMA Source Address Increment Register 3
0xB8E8	64	DM1DAIR3	DMA Destination Address Increment Register 3
0xB8F0	64	DM1CCR3	DMA Channel Control Register 3
0xB8F8	64	DM1CSR3	DMA Channel Status Register 3
0xB948	64	DM1MFDR	DMA Memory Fill Data Register
0xB950	64	DM1MCR	DMA Master Control Register

8.4.1 DMA Master Control Register (DM0MCR, DM1MCR)

Offset address: DMAC0 0xB150, DMAC1 0xB950

This register controls the entire DMA Controller.



Bit	Mnemonic	Field Name	Description	Read/Write
63:32		Reserved		—
31:28	EIS[3:0]	Error Interrupt Status	Error Interrupt Status [3:0] (Default: 0x0) These four bits indicate the error interrupt status of each channel. EIS[n] corresponds to channel *n*. 1: There is an error interrupt in the corresponding channel. 0: There is no error interrupt in the corresponding channel.	R
27:24	DIS[3:0]	Normal Completion Interrupt Status	Done Interrupt Status [3:0] (Default: 0x0) These four bits indicate the transfer completion (transfer complete or chain ended) interrupt status of each channel. DIS[n] corresponds to channel *n*. 1: There is a transfer completion interrupt in the corresponding channel. 0: There is no transfer completion interrupt in the corresponding channel.	R
23:21		Reserved		—
20:14	FIFVC	FIFO Valid Entry Count	FIFO Valid Entry Count (Default: 0000000) These read only bits indicate the byte count of data that were written to FIFO but not read out from the FIFO.	R
13:11	FIFWP	FIFO Write Pointer	FIFO Write Pointer (Default: 000) These read only bits indicate the next write position in FIFO. This is a diagnostic function.	R
10:8	FIFRP	FIFO Read Pointer	FIFO Read Pointer (Default: 000) These read only bits indicate the next read position in FIFO. This is a diagnostic function.	R
7	RSFIF	Reset FIFO	Reset FIFO (Default: 0) This bit is used for resetting FIFO. When this bit is set to “1”, the FIFO read pointer, FIFO write pointer and FIFO valid entry count are initialized to “0”. If an error occurs during DMA transfer, use this bit when data remains in the FIFO (when the FIFO Valid entry Count Field is not “0”) to initialize the FIFO.	R/W

Figure 8.4.1 DMA Master Control Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
6:3	FIFUM[3:0]	FIFO Use Enable [3:0]	FIFO Use Enable [3:0] (Default: 0x0) Each channel specifies whether to use 8-double word FIFO in Dual Address transfer. FIFUM[n] corresponds to channel *n*. Refer to “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information.	R/W
2		Reserved		—
1	RRPT	Round Robin Priority	Round Robin Priority (Default: 0) Specifies the method for determining priority among channels. 1: Round Robin method. Priority of the last channel used is the lowest, and the next previous channel has the next lowest priority. Round robin is in the order Channel 0 > Channel 1 > Channel > Channel 3. 0: Fixed Priority. Priority is fixed in the order Channel 0 > Channel 1 > Channel 2 > Channel 3.	R/W
0	MSTEN	Master Enable	Master Enable (Default: 0) This bit enables the DMA Controller. 1: Enable 0: Disable Note: If the entire DMA Controller is disabled, then all internal logic including the Bus Interface Logic and State Machine are reset.	R/W

Figure 8.4.1 DMA Master Control Register (2/2)

8.4.2 DMA Channel Control Register (DM0CCRn, DM1CCRn)

Offset address: DMAC0 0xB030 (ch. 0) / 0xB070 (ch. 1) / 0xB0B0 (ch. 2) / 0xB0F0 (ch. 3)

DMAC1 0xB830 (ch. 0) / 0xB870 (ch.1) / 0xB8B0 (ch. 2) / 0xB8F0 (ch. 3)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Reserved															
: Type															
: Initial value															

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Reserved															
: Type															
: Initial value															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		IMMCHN	USEXFSZ	LE	DBINH	SBINH	CHRST	RVBYTE	ACKPOL	REQPL	EGREQ	CHDN	DNCTL		EXTRQ
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W
		0	0	—	0	0	1	0	0	0	0	0	00		0

: Type
: Initial value

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STLTIME/INTRQD			INTENE	INTENC	INTENT	CHNEN	XFACT	Reserved		SMPCHN	XFSZ			MEMIO	SNGAD
R/W			R/W	R/W	R/W	R	R/W			R/W	R/W			R/W	R/W
000			0	0	0	0	0			0	000			0	0

: Type
: Initial value

Bit	Mnemonic	Field Name	Description	Read/Write
63:32		Reserved		—
29	IMMCHN	Immediate Chain	Immediate Chain (Default: 0) Always set this bit to “1”.	R/W
28	USEXFSZ	Transfer Set Size Mode	Use Transfer Set Size (Default: 0) Selects the DMA channel operation mode during Burst DMA transfer. Refer to “8.3.7.2 Burst Transfer During Single Address Transfer” and “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information. 1: The DMA Controller always transfers the amount of data set in DMCCRn.XFSZ for each bus operation. Since alignment to the boundary of the DMCCRn.XFSZ in the address is not forced when in this mode, transfers that exceed 32-double-word boundaries are divided into two operations. 0: The DMA Controller calculates the transfer size so the address set in DMSARn and DMDARn (only during Dual Address transfer) can be aligned to the boundary of the size set in DMCCRn.XFSZ, then transfers data according to that size. Note: In Dual Address Transfer mode, programming this bit to 1 is valid only when both the contents of the DMSARn and the DMDARn are on doubleword boundaries *and* the contents of the DMCNTRn is a multiple of eight bytes.	R/W
27	LE	Little Endian	Little Endian (Default: value that is the opposite of the G-Bus Endian (CCFG.ENDIAN) This bit sets the Endian of the channel. Please use the default value as is. 1: Channel operates in the Little Endian mode 0: Channel operates in the Big Endian mode	R/W

Figure 8.4.2 DMA Channel Control Register (1/4)

Bit	Mnemonic	Field Name	Description	Read/Write
26	DBINH	Destination Burst Inhibit	Destination Burst Inhibit (Default: 0) During Dual Address transfer, this bit sets whether to perform Burst transfer or Single transfer on a Write cycle to the address set from FIFO to DMDARn when Burst transfer is set by DMCCRn.XFSZ. Refer to “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information. The settings of this bit have no effect during Single Address transfers. 1: Multiple Single transfers are executed. 0: Burst transfer is executed.	R/W
25	SBINH	Source Burst Inhibit	Source Burst Inhibit (Default: 0) During Dual Address transfer, this bit sets whether to perform Burst transfer or Single transfer on a Read cycle to the FIFO from the address set to DMSARn when Burst transfer is set by DMCCRn.XFSZ. Refer to “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information. The settings of this bit have no effect during Single Address transfers. 1: Multiple Single transfers are executed. 0: Burst transfer is executed.	R/W
24	CHRST	Channel Reset	Channel Reset (Default: 1) This bit is used fo initializing channels. The DMCCRn.XFACT, DMCCRn.CHNEN, and DMCSRn bits are all cleared. In addition, all channel logic and interrupts from channels are cleared and bus ownership requests to the DMA Channel Arbiter are also reset. The software must clear this bit before operating a channel. 1: Reset channel 0: Enable channel	R/W
23	REVBYTE	Reverse Byte	Reverse Bytes (Default: 0) This bit specifies whether to reverse the byte order during a Dual Address transfer when the Transfer Setting Size field (DMCCRn.XFSZ) setting is 8 bytes or more. Refer to “8.3.8.3 Double Word Byte Swapping” for more information. 1: Reverses the byte order. 0: Does not reverse the byte order.	R/W
22	ACKPOL	Acknowledge Polarity	Acknowledge Polarity (Default: 0) Specifies the polarity of the DMAACK[n] signal. 1: Asserts when the DMAACK[n] signal is High 0: Asserts when the DMAACK[n] signal is Low	R/W
21	REQPL	Request Polarity	Request Polarity (Default: 0) Specifies the polarity of the DMAREQ[n] signal. 1: Asserts when the DMAREQ[n] signal is High. 0: Asserts when the DMAREQ[n] signal is Low.	R/W
20	EGREQ	Edge Request	Edge Request (Default: 0) Specifies the method for detecting DMA requests by the DMAREQ[n] signal. 1: DMAREQ[n] signal is Edge Detect. 0: DMAREQ[n] signal is Level Detect.	R/W
19	CHDN	Chain Complete	Chain Done (Default: 0) Selects control by the DMADONE* signal. See “8.3.3.4 DMA Controller” for more information. 1: Assertion of the DMADONE* signal controls the overall Chain DMA transfer. 0: Assertion of the DMADONE* signal controls DMA transfer according to the DMA Channel Register setting at that time.	R/W
18:17	DNCTL	DONE Control	Done Control (Default: 00) Specifies the input/output mode of the DMADONE* signal. Refer to “8.3.3.4 DMADONE* Signal” for more information. 00: DMADONE* signal becomes the input signal, but input is ignored. 01: DMADONE* signal becomes the input signal. 10: DMADONE* signal becomes the output signal. 11: DMADONE* signal becomes the open drain input/output signal.	R/W

Figure 8.4.2 DMA Channel Control Register (2/4)

Bit	Mnemonic	Field Name	Description	Read/Write
16	EXTRQ	External Request	External Request (Default: 0) Sets the Request Transfer mode. 1: I/O DMA transfer mode This bit is used by the External I/O DMA Transfer mode and the Internal I/O DMA Transfer mode. A channel requests internal bus ownership when the I/O device asserts the DMA request signal. 0: Memory Transfer mode This bit is used by the Memory-Memory Copy Transfer mode and the Memory Fill Transfer mode. A channel requests internal bus ownership when the value of DMCSRn.WAITC becomes “0”.	R/W
15:13	STLTIME / INTRQD	Transfer Stall Detection Interval/Internal Request Delay	• When in the I/O DMA Transfer mode (DMCCRn.EXTRQ is “1”) Stalled Transfer Detect Time (Default: 000) Sets the detection interval for a lack of bus ownership. If this channel *n* releases bus ownership then the interval it does not have ownership exceeds the clock count set by this field, then DMCSRn.STLXFER is set to “1”. Refer to “8.3.13 Transfer Stall Detection Function” for more information. 000: Does not detect stalled transfers. 001: Sets 960 (15 × 64) clocks as the detection interval 010: Sets 4032 (63 × 64) clocks as the detection interval 011: Sets 16320 (255 × 64) clocks as the detection interval 100: Sets 65472 (1023 × 64) clocks as the detection interval 101: Sets 262080 (4095 × 64) clocks as the detection interval 110: Sets 1048512 (16383 × 64) clocks as the detection interval 111: Sets 4194240 (65535 × 64) clocks as the detection interval • When in the Memory Transfer mode (DMCCRn.EXTRQ is “0”) Internal Request Delay (Default: 000) Sets the delay time from when bus ownership is released to the next bus ownership request. Bus ownership is released, the set delay time elapses, then a bus ownership request is generated from the channel. 000: Always requests bus ownership when this channel is active. (Bus ownership is released after bus operation ends) 001: Set 16 clocks as the delay time 010: Set 32 clocks as the delay time 011: Set 64 clocks as the delay time 100: Set 128 clocks as the delay time 101: Set 256 clocks as the delay time 110: Set 512 clocks as the delay time 111: Set 1024 clocks as the delay time	R/W
12	INTENE	Error Interrupt Enable	Interrupt Enable on Error (Default: 0) Enables interrupts when the Error End bit (DMCSRn.ABCHC) or the Transfer Stall Detection bit (DMCSRn.STLXFER) is set. 1: Generates interrupts. 0: Does not generate interrupts.	R/W
11	INTENC	Chain End Interrupt Enable	Interrupt Enable on Chain Done (Default: 0) This bit enables interrupts when the Chain End bit (DMCSRn.NCHNC) is set. 1: Generate interrupts. 0: Do not generate interrupts.	R/W
10	INTENT	Transfer End Interrupt Enable	Interrupt Enable on Transfer Done (Default: 0) This bit enables interrupts when the Transfer End bit (DMCSRn.NTRNFC) is set. 1: Generate interrupts. 0: Do not generate interrupts.	R/W

Figure 8.4.2 DMA Channel Control Register (3/4)

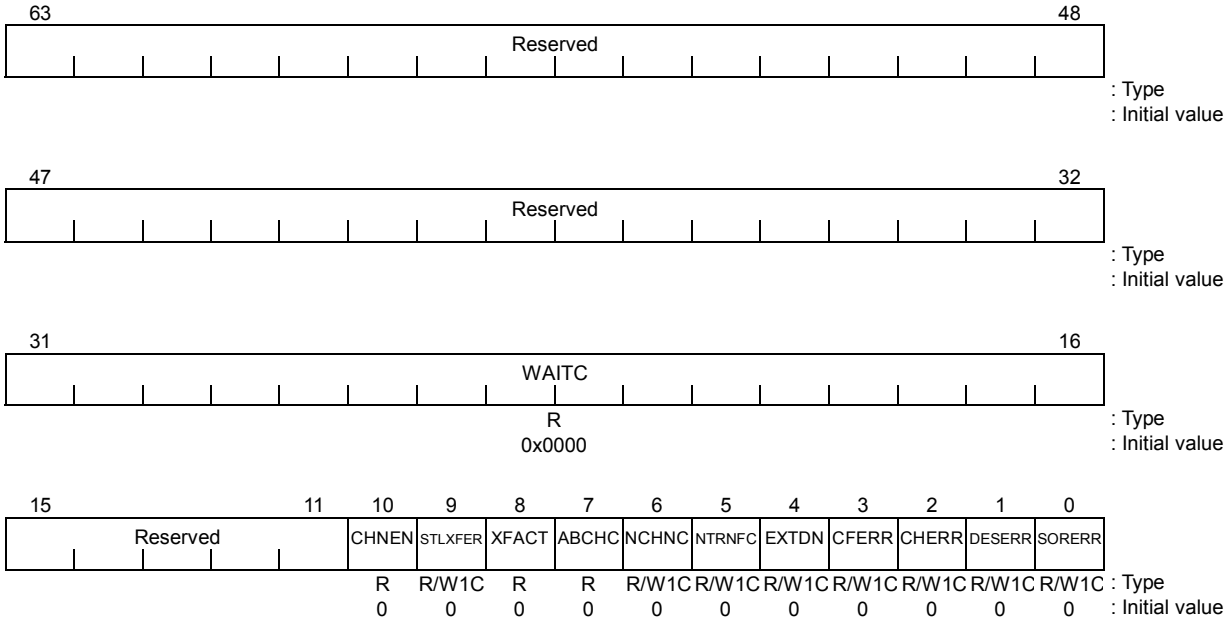
Bit	Mnemonic	Field Name	Description	Read/Write
9	CHNEN	Chain Enable	Chain Enable (Default: 0) This bit indicates whether Chain operation is being performed. Read Only. This bit is cleared when either the Master Enable bit (DMMCR.MSTEN) is cleared or the Channel Reset bit (DMCCRn.CHRST) is set. This bit is set if a value other than "0" is set when the CPU writes to the DMA Chain Address Register (DMCHARn) or when a Chain transfer writes DMA Command Descriptor. This bit is then cleared when "0" is set to the DMA Chain Address Register (DMCHARn). 1: If transfer completes due to the current DMA Channel Register setting, a DMA Command Descriptor is loaded in the DMA Channel Register from the specified DMA Chain Address Register (DMCHARn) address, then DMA transfer continues. 0: Further transfer does not start even if transfer completes due to the current DMA Channel Register setting.	R
8	XFACT	Transfer Active	Transfer Active (Default: 0) DMA transfer is performed according to the DMA Channel Register setting when this bit is set. This bit is automatically set when a value other than "0" is set in the DMA Chain Address Register (DMCHARn). DMA transfer is then initiated. This bit is automatically cleared either when DMA transfer ends normally it is stopped due to an error. 1: Perform DMA transfer. 0: Do not perform DMA transfer.	R/W
7:6		Reserved		—
5	SMPCHN	Simple Chain	Simple Chain (Default: 0) This bit selects the DMA Channel Register that loads data from DMA Command Descriptors during Chain DMA transfer. 1: Data is only loaded to the four following DMA Channel Registers: the Chain Address Register (DMCHARn), the Source Address Register (DMSARn), the Destination Address Register (DMDARn), and the Count Register (DMCNTRn). 0: Data is loaded to all eight DMA Channel Registers.	R/W
4:2	XFSZ	Transfer Set Size	Transfer Set Size (Default: 000) These bits set the transfer data size of each bus operation in the internal bus. When the transfer set size is set to four double words or greater, the data size actually transferred during a single bus operation does not always match the transfer set size. Refer to "8.3.7.2 Burst Transfer During Single Address Transfer" and "8.3.8.2 Burst Transfer During Dual Address Transfer" for more information. 000: 1 byte 001: 2 byte 010: 4 byte 011: 8 bytes (1double word) 100: 4 double words 101: 8 double words 110: 16 double words (Single Address transfer only) 111: 32 double words (Single Address transfer only)	R/W
1	MEMIO	Memory to I/O	Memory to I/O (Default: 0) This bit specifies the transfer direction during Single Address transfer (DMCCRn.SNGAD = 1). Clear this bit when in the Memory Fill Transfer mode. The setting of this bit is ignored when Dual Address transfer is set (DMCCRn.SNGAD = 0). 1: From memory to I/O 0: From I/O to memory	R/W
0	SNGAD	Single Address	Single Address (Default: 0) This bit specifies whether the transfer method is Single Address transfer or Dual Address transfer. 1: Single Address transfer 0: Dual Address transfer	R/W

Figure 8.4.2 DMA Channel Control Register (4/4)

8.4.3 DMA Channel Status Register (DM0CSRn, DM1CSRn))

Offset Address: DMAC0 0xB038 (ch. 0) / 0xB078 (ch. 1) / 0xB0B8 (ch. 2) / 0xB0F8 (ch. 3)

DMAC1 0xB838 (ch. 0) / 0xB878 (ch. 1) / 0xB8B8 (ch. 2) / 0xB8F8 (ch. 3)



Bit	Mnemonic	Field Name	Description	Read/Write
63:32		Reserved		—
31:16	WAITC	Wait Counter	Wait Counter (Default: 0x0000) This is a diagnostic function. • I/O DMA transfer mode (DMCCRn.EXTRQ = "1") This counter is decremented by 1 at each 64 G-Bus cycles. After channel *n* releases bus ownership, this counter sets the default (the value that is the detection interval clock cycle count set by the Transfer Stall Detection Interval field (DMCCRn.STLTIME) divided by 64). The Transfer Stall Detect bit (DMCSRn.STLXFER) is set when the interval during which bus ownership is not held reaches the set clock cycle. The counter is reset to the default and stops counting. Clearing the Transfer Stall Detect bit (DMCSRn.STLXFER) resumes the count and starts stall detection. • Memory transfer mode (DMCCRn.EXTRQ = "0") This counter is decremented by 1 at each G-Bus cycle. After bus ownership is released, the counter is set to the delay clock cycle count set by the Internal Request Delay field (DMCCRn.INTRQD). When the counter reaches "0" the count stops and channel *n* requests bus ownership.	R
15:11		Reserved		—
10	CHNEN	Chain Enable	Chain Enable (Default: 0) This value is a copy of the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn).	R
9	STLXFER	Transfer Stall Detect	Stalled Transfer Detect (Default: 0) This bit indicates whether the interval during which bus ownership is not held exceeds the value set by the Transfer Stall Detect Interval field (DMCCRn.STLTIME) after bus ownership is released when in the I/O DMA transfer mode. 1: Indicates that the interval during which bus ownership was not held exceeds the DMCCRn.STLTIME setting. 0: The interval during which bus ownership was not held did not exceed the setting since this bit was last cleared.	R/W1C

Figure 8.4.3 DMA Channel Status Register (1/2)

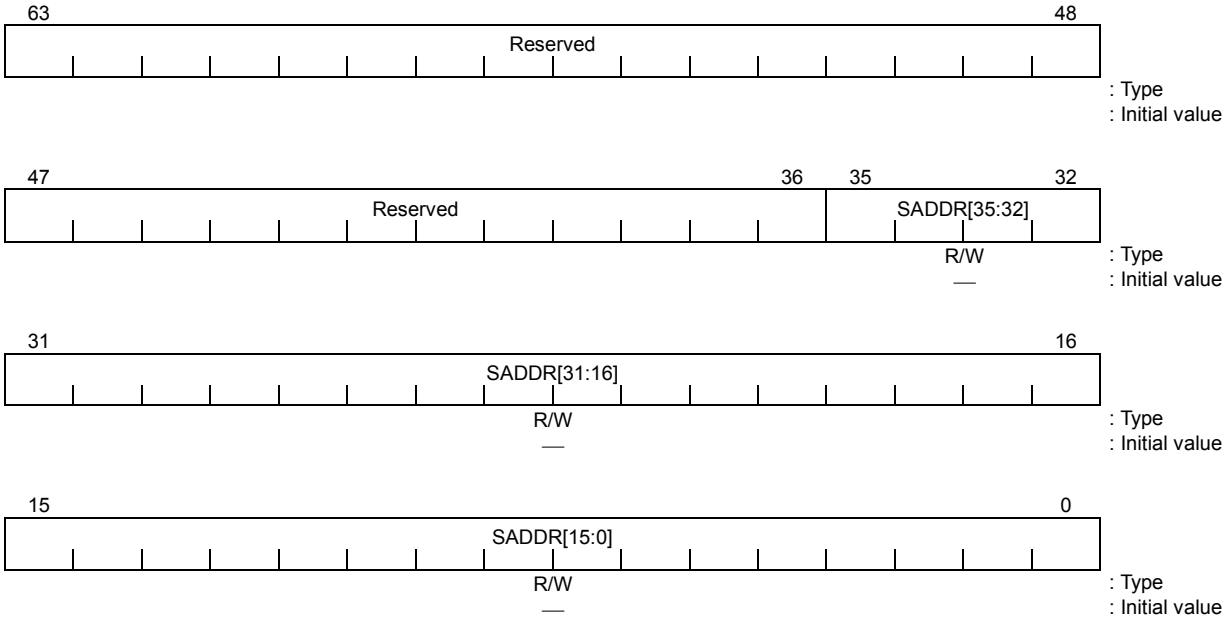
Bit	Mnemonic	Field Name	Description	Read/Write
8	XFACT	Transfer Active	Transfer Active (Default: 0) This value is a copy of the Transfer Active bit (XFACT) of the DMA Channel Control Register (DMCCRn).	R
7	ABCHC	Error Complete	Error Completion (Default: 0) This bit indicates whether an error occurred during DMA transfer. This bit indicates the logical sum of the four error bits (CFERR, CHERR, DESERR, SORERR) in DMCSRn[3:0]. 1: DMA transfer ends due to an error. 0: No error occurred since this bit was last cleared.	R
6	NCHNC	Chain Complete	Normal Chain Completion (Default: 0) When performing chain DMA transfer, This bit indicates whether all DMA data transfers in the DMA Descriptor chain are complete. 1: All DMA data transfers in the DMA Descriptor chain ended normally. Or, DMA transfer that did not use a DMA Descriptor chain ended normally. 0: DMA transfer has not ended normally since this bit was last cleared.	R/W1C
5	NTRNFC	Transfer Complete	Normal Transfer Completion (Default: 0) This bit indicates whether DMA transfer ended according to the current DMA Channel Register setting. 1: DMA transfer ended normally. 0: DMA transfer has not ended since this bit was last cleared.	R/W1C
4	EXTDN	External DONE Asserted	External Done Asserted (Default: 0) This bit indicates whether an external I/O device asserted the DMADONE* signal. When the DMADONE* signal is set to bidirectional, this bit is also set when the TX4938 asserts the DMADONE* signal. 1: DMADONE* signal was asserted. 0: DMADONE* signal was not asserted.	R/W1C
3	CFERR	Configuration Error	Configuration Error (Default: 0) Indicates whether an illegal register setting was made. 1: There was a configuration error. 0: There was no configuration error.	R/W1C
2	CHERR	Chain Bus Error	Chain Bus Error (Default: 0) This bit indicates whether a bus error occurred while reading a DMA Command Descriptor. 1: Bus error occurred. 0: No bus error occurred.	R/W1C
1	DESERR	Destination Error	Destination Bus Error (Default: 0) This bit indicates whether a bus error occurred during a destination bus Write operation (a Write to a set DMDARn address). 1: Bus error occurred. 0: No bus error occurred.	R/W1C
0	SORERR	Source Bus Error	Source Bus Error (Default: 0) This bit indicates whether a bus error occurred during either a source bus Read or Write operation (A Read or Write to a set DMSARn address). 1: Bus error occurred. 0: No bus error occurred.	R/W1C

Figure 8.4.3 DMA Channel Status Register (2/2)

8.4.4 DMA Source Address Register (DM0SARn, DM1SARn)

Offset Address: DMAC0 0xB008 (ch. 0) / 0xB048 (ch. 1) / 0xB088 (ch. 2) / 0xB0C8 (ch. 3)

DMAC1 0xB808 (ch. 0) / 0xB848 (ch. 1) / 0xB888 (ch. 2) / 0xB8C8 (ch. 3)

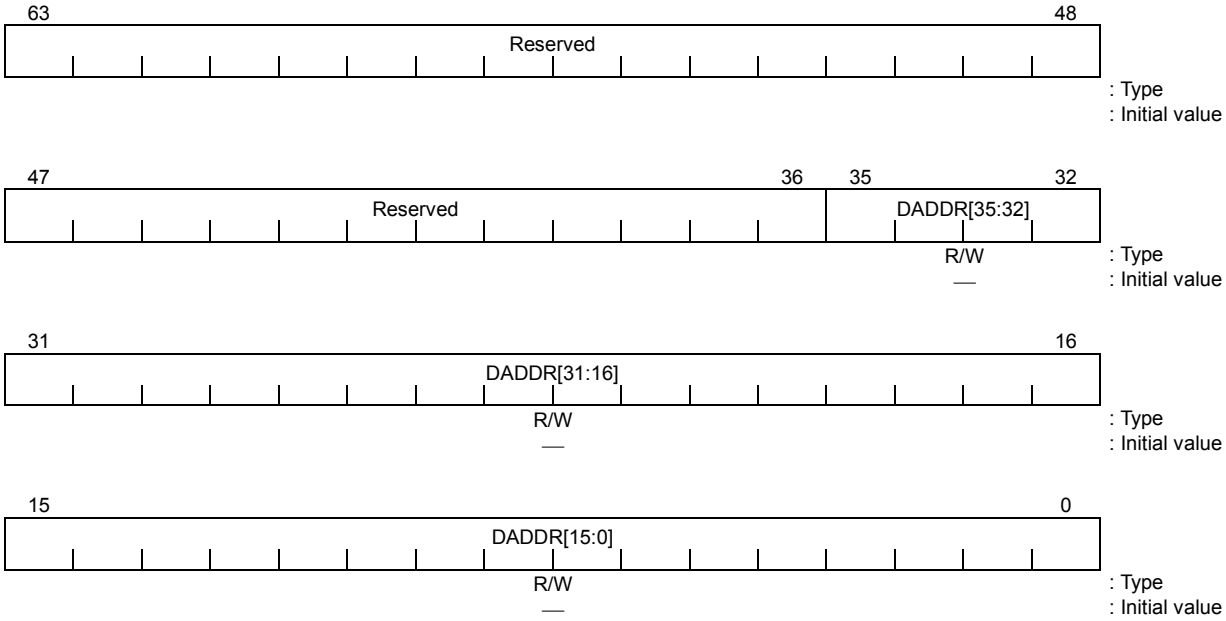


Bits	Mnemonic	Field Name	Description	Read/Write
63:36	—	Reserved		—
35:0	SADDR	Source Address	Source Address (Default: Undefined) This field sets the physical address of the transfer source during Dual Address transfer. This field sets the physical address of memory access during Single Address transfer. This field is used for either Memory-to-I/O or I/O-to-Memory transfers. Refer to “8.3.7.1 Channel Register Settings During Single Address Transfer” and “8.3.8.1 Channel Register Settings During Dual Address Transfer” for more information. During Burst transfer, the value changes once for each bus operation only by the size that was transferred. During Single transfer, the value only changes by the value specified by the DMA Source Address Increment Register (DMSAIRn).	R/W

Figure 8.4.4 DMA Source Address Register

8.4.5 DMA Destination Address Register (DM0DARn, DM1DARn)

Offset Address: DMAC0 0xB010 (ch. 0) / 0xB050 (ch. 1) / 0xB090 (ch. 2) / 0xB0D0 (ch. 3)
DMAC1 0xB810 (ch. 0) / 0xB850 (ch. 1) / 0xB890 (ch. 2) / 0xB8D0 (ch. 3)



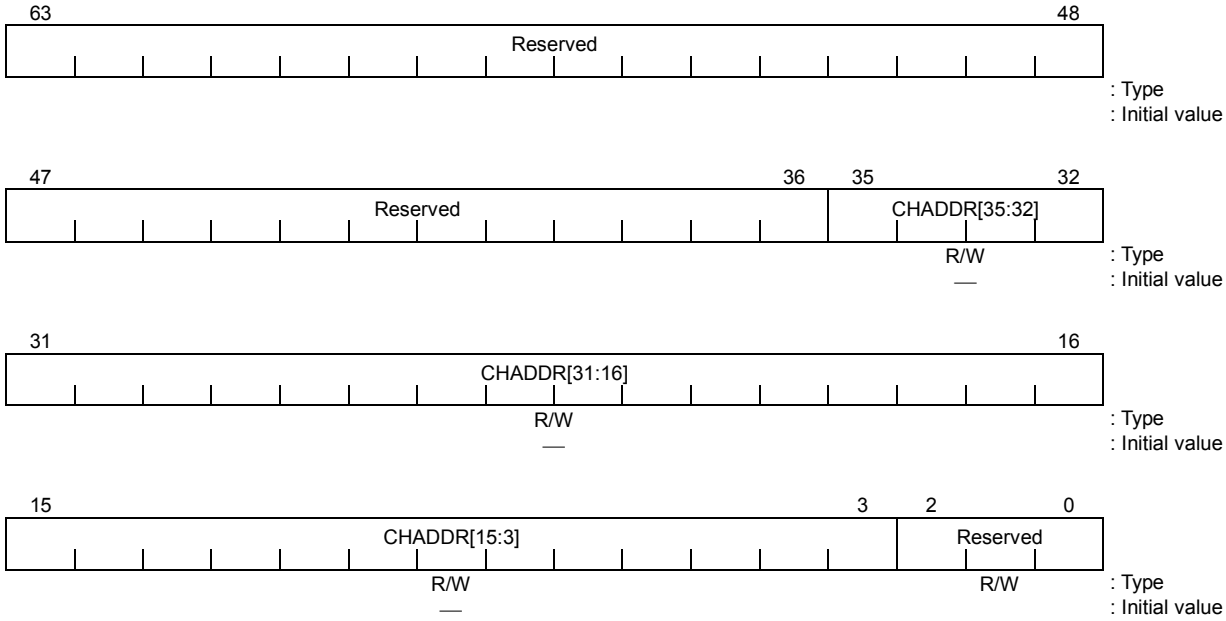
Bit	Mnemonic	Field Name	Description	Read/Write
63:36		Reserved		—
35:0	DADDR	Destination Address	Destination Address (Default: undefined) This register sets the physical address of the transfer destination during Dual Address transfer. This register is ignored during Single Address transfer. Refer to “8.3.8.1 Channel Register Settings During Dual Address Transfer” for more information. During Burst transfer, the value changes only by the size of data transferred during each single bus operation. During Single transfer, the value only changes by the value specified by the DMA Destination Address Increment Register (DMDAIRn).	R/W

Figure 8.4.5 DMA Destination Address Register

8.4.6 DMA Chain Address Register (DM0CHARn, DM1CHARn)

Offset Address: DMAC0 0xB000 (ch. 0) / 0xB040 (ch. 1) / 0xB080 (ch. 2) / 0xB0C0 (ch. 3)

DMAC1 0xB800 (ch. 0) / 0xB840 (ch. 1) / 0xB880 (ch. 2) / 0xB8C0 (ch. 3)



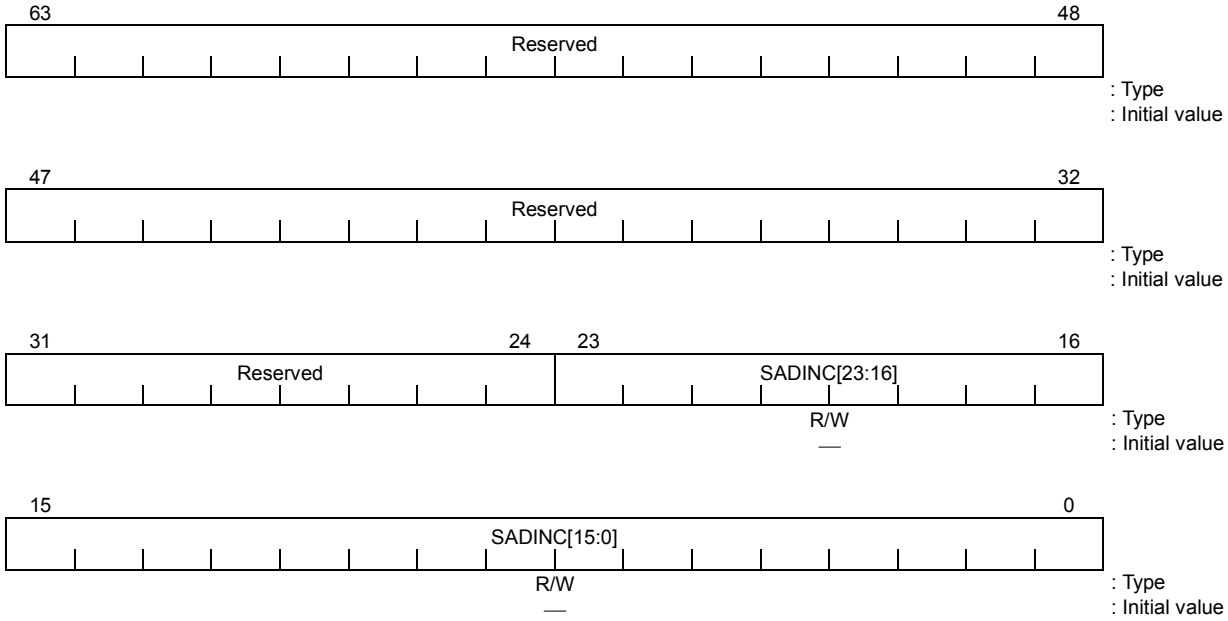
Bit	Mnemonic	Field Name	Description	Read/Write
63:36		Reserved		—
35:3	CHADDR	Chain Address	Chain Address (Default: undefined) When Chain DMA transfer is executed, this register sets the physical address of the next DMA Command Descriptor to be read. If DMA transfer according to the current Channel Register setting ends and the Chain Enable bit (DMCCRn.CHNEN) is set, then the DMA Command Descriptor is loaded in the Channel Register starting from the address indicated by this register. When a value other than “0” is set in this register, the Chain Enable bit (DMCCRn.CHNEN) and the Transfer Active bit (DMCCRn.XFACT) are set. When “0” is set in this register, only the Chain Enable bit (DMCCRn.CHNEN) is cleared. When the Chain Address field value reads a DMA Command Descriptor of 0, the value of this register is not updated and the value before that one (address of the Data Command Descriptor when the value of the Chain Address field being read was “0”) is held.	R/W
2:0		Reserved		R/W

Figure 8.4.6 DMA Chain Address Register

8.4.7 DMA Source Address Increment Register (DM0SAIRn, DM1SAIRn)

Offset Address: DMAC0 0xB020 (ch. 0) / 0xB060 (ch. 1) / 0xB0A0 (ch. 2) / 0xB0E0 (ch. 3)

DMAC1 0xB820 (ch. 0) / 0xB860 (ch. 1) / 0xB8A0 (ch. 2) / 0xB8E0 (ch. 3)



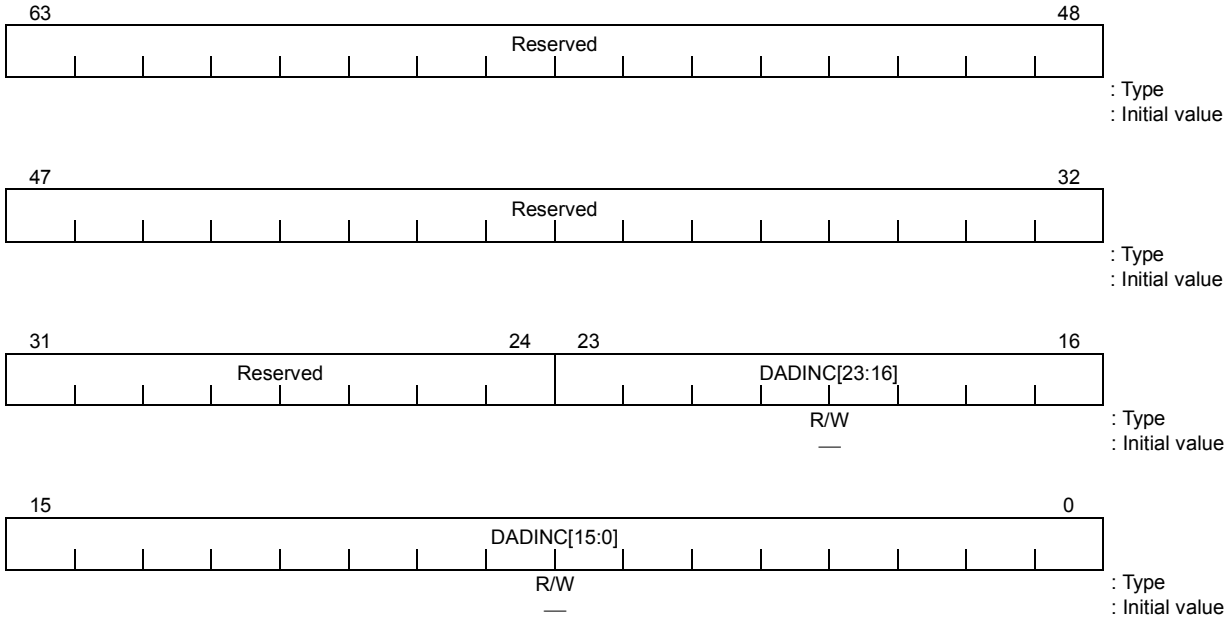
Bit	Mnemonic	Field Name	Description	Read/Write
63:24		Reserved		—
23:0	SADINC	Source Address Increment	Source Address Increment (Default: undefined) This field sets the increase/decrease value of the DMA Source Address Register (DMSARn). This value is a 24-bit two's complement and indicates a byte count. Refer to "8.3.7.1 Channel Register Settings During Single Address Transfer" and "8.3.8.1 Channel Register Settings During Dual Address Transfer" for more information.	R/W

Figure 8.4.7 DMA Source Address Increment Register

8.4.8 DMA Destination Address Increment Register (DM0DAIRn, DM1DAIRn)

Offset Address: DMAC0 0xB028 (ch. 0) / 0xB068 (ch. 1) / 0xB0A8 (ch. 2) / 0xB0E8 (ch. 3)

DMAC1 0xB828 (ch. 0) / 0xB868 (ch. 1) / 0xB8A8 (ch. 2) / 0xB8E8 (ch. 3)



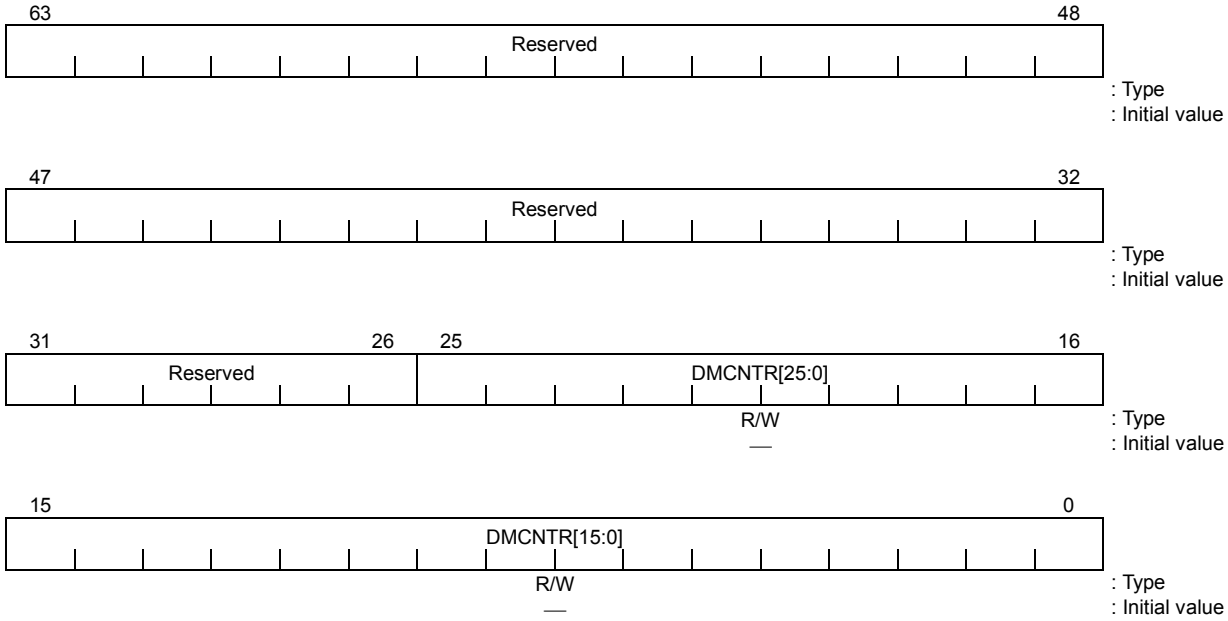
Bit	Mnemonic	Field Name	Description	Read/Write
63:24		Reserved		—
23:0	DADINC	Destination Address Increment	Destination Address Increment (Default: undefined) This field sets the increase/decrease value of the DMA Destination Address Register (DMDARn). This value is a 24-bit two's complement and indicates a byte count. Refer to "8.3.8.1 Channel Register Settings During Dual Address Transfer" for more information.	R/W

Figure 8.4.8 DMA Destination Address Increment Register

8.4.9 DMA Count Register (DM0CNTRn, DM1CNTRn)

Offset Address: DMAC0 0xB018 (ch. 0) / 0xB058 (ch. 1) / 0xB098 (ch. 2) / 0xB0D8 (ch. 3)

DMAC1 0xB818 (ch. 0) / 0xB858 (ch. 1) / 0xB898 (ch. 2) / 0xB8D8 (ch. 3)

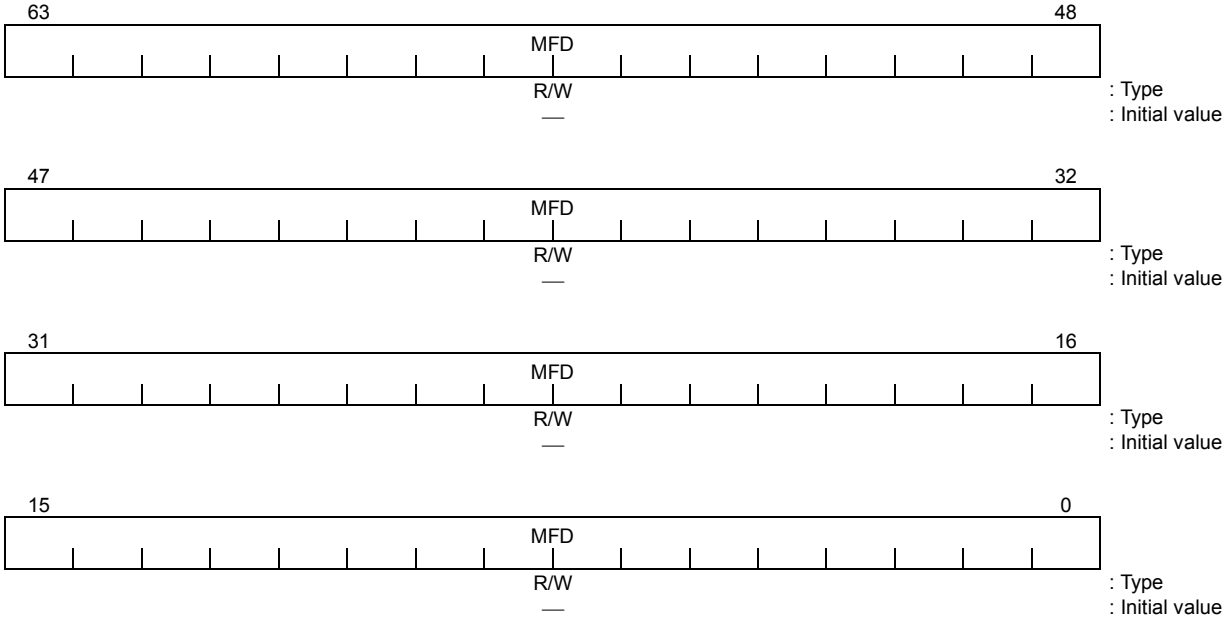


Bit	Mnemonic	Field Name	Description	Read/Write
63:26		Reserved		—
25:0	DMCNTR	Count	Count Register (Default: undefined) This register sets the byte count that is transferred by the DMA Channel Register setting. The value is a 26-bit unsigned data that is decremented only by the size of the data transferred during a single bus operation. Refer to “8.3.7.1 Channel Register Settings During Single Address Transfer” and “8.3.8.1 Channel Register Settings During Dual Address Transfer” for more information.	R/W

Figure 8.4.9 DMA Count Register

8.4.10 DMA Memory Fiill Data Register (DM0MFDR, DM1MFDR)

Offset Address: DMAC0 0xB148, DMAC1 0xB948



Bit	Mnemonic	Field Name	Description	Read/Write
63:0	MFD	Memory Fill Data	Memory Fill Data (Default: undefined) This register, which stores double-word data written to memory when in the Memory Fill Transfer mode, is shared between all channels.	R/W

Figure 8.4.10 DMA Memory Fill Data Register

8.5 Timing Diagrams

This section contains timing diagrams for the external I/O DMA transfer mode. The DMAREQ[n] signals and DMAACK[n] signals in the timing diagrams are set to Low Active.

8.5.1 Single Address Single Transfer from Memory to I/O (32-bit ROM)

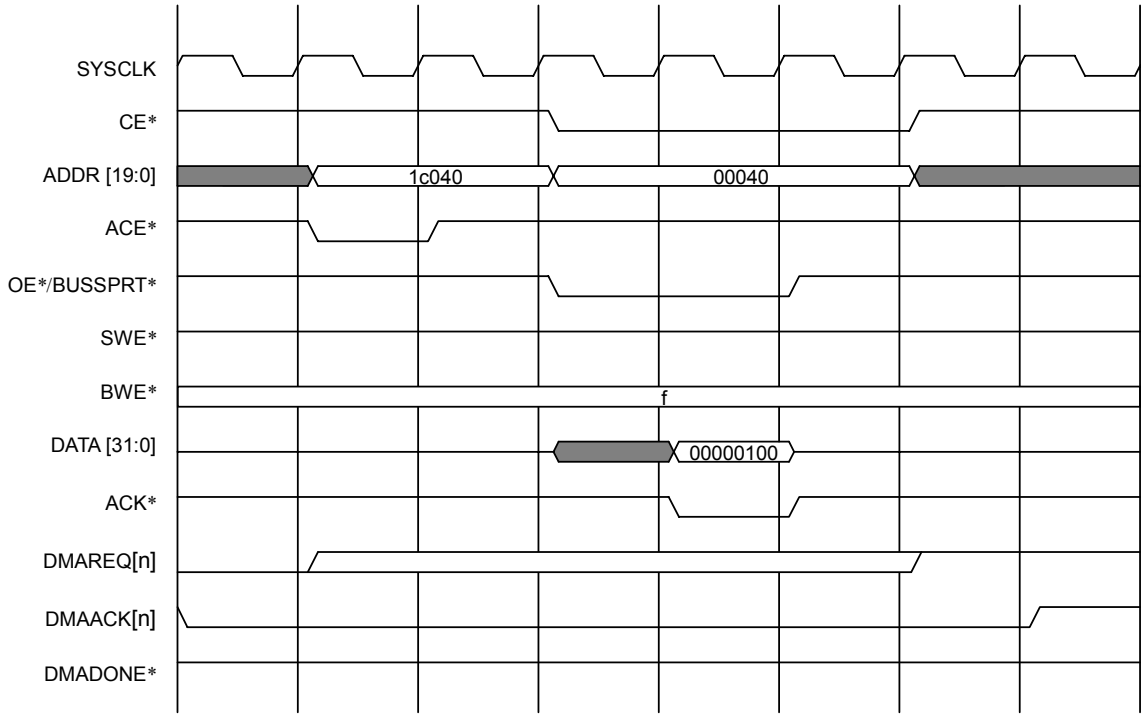


Figure 8.5.1 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 32-bit ROM)

8.5.2 Single Address Single Transfer from Memory to I/O (16-bit ROM)

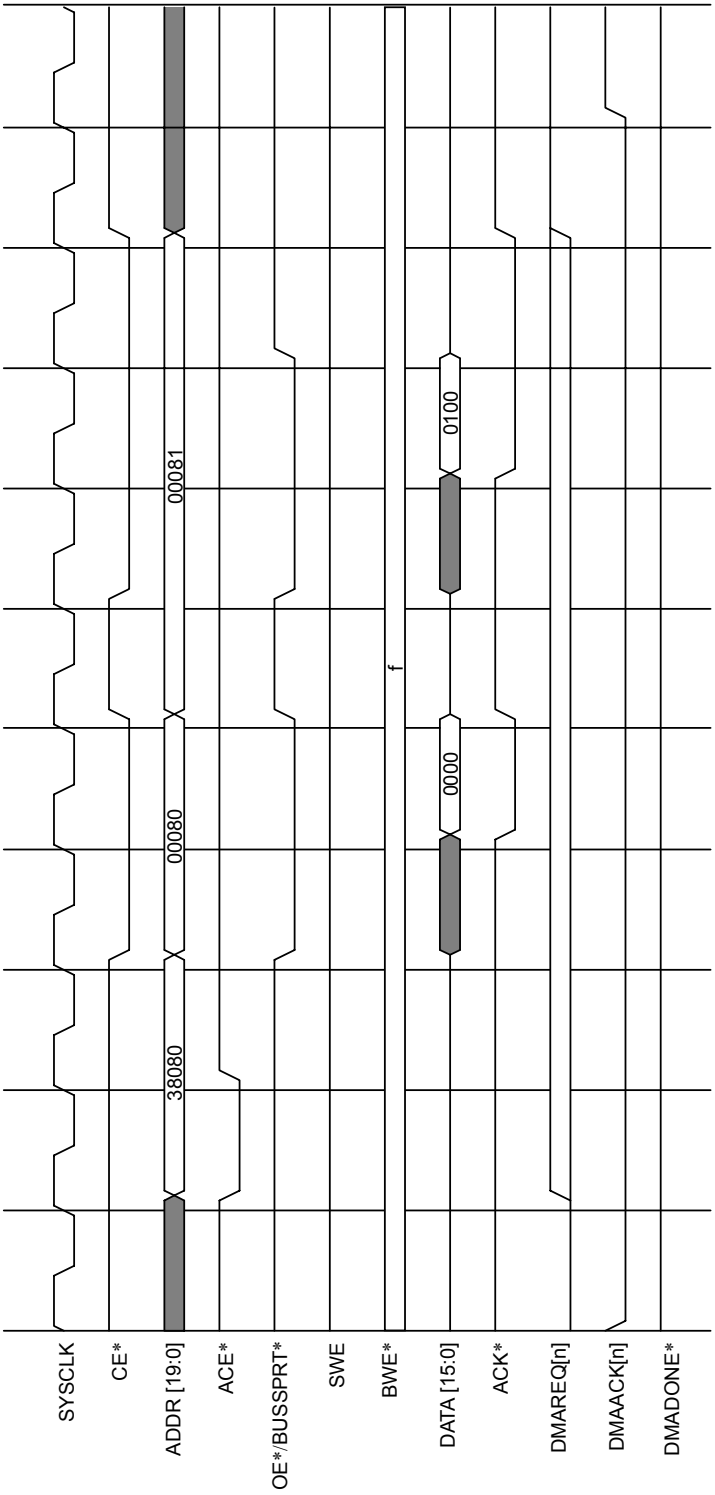


Figure 8.5.2 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 16-bit ROM)

8.5.3 Single Address Single Transfer from I/O to Memory (32-bit SRAM)

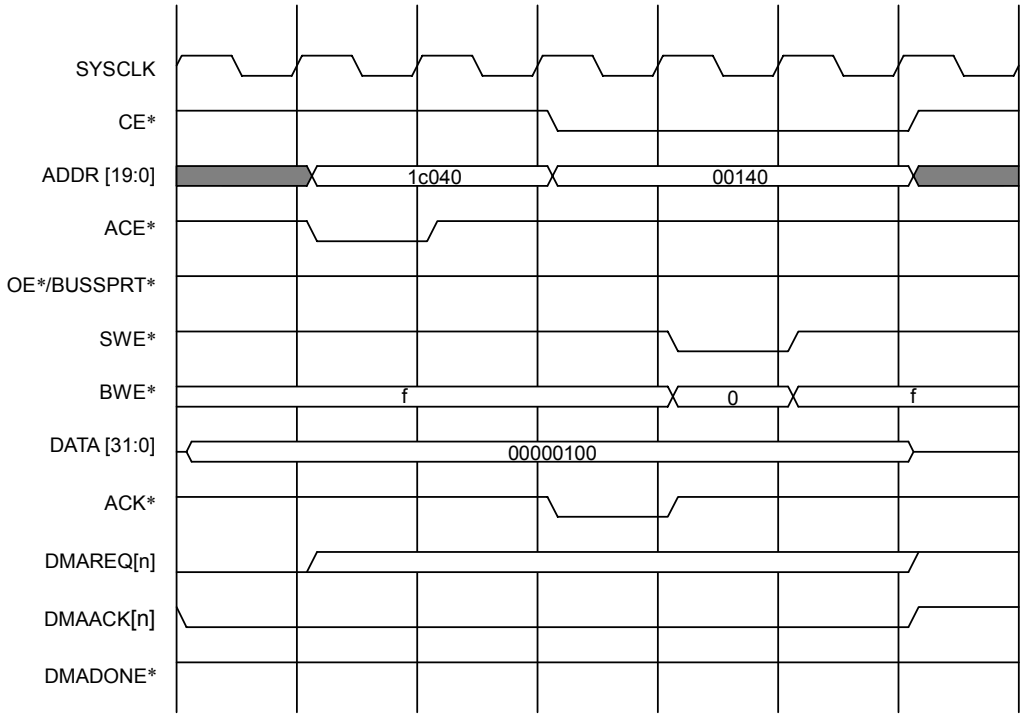


Figure 8.5.3 Single Address Single Transfer from I/O to Memory
(Single Write of 32-bit Data to 32-bit SRAM)

8.5.4 Single Address Burst Transfer from Memory to I/O (32-bit ROM)

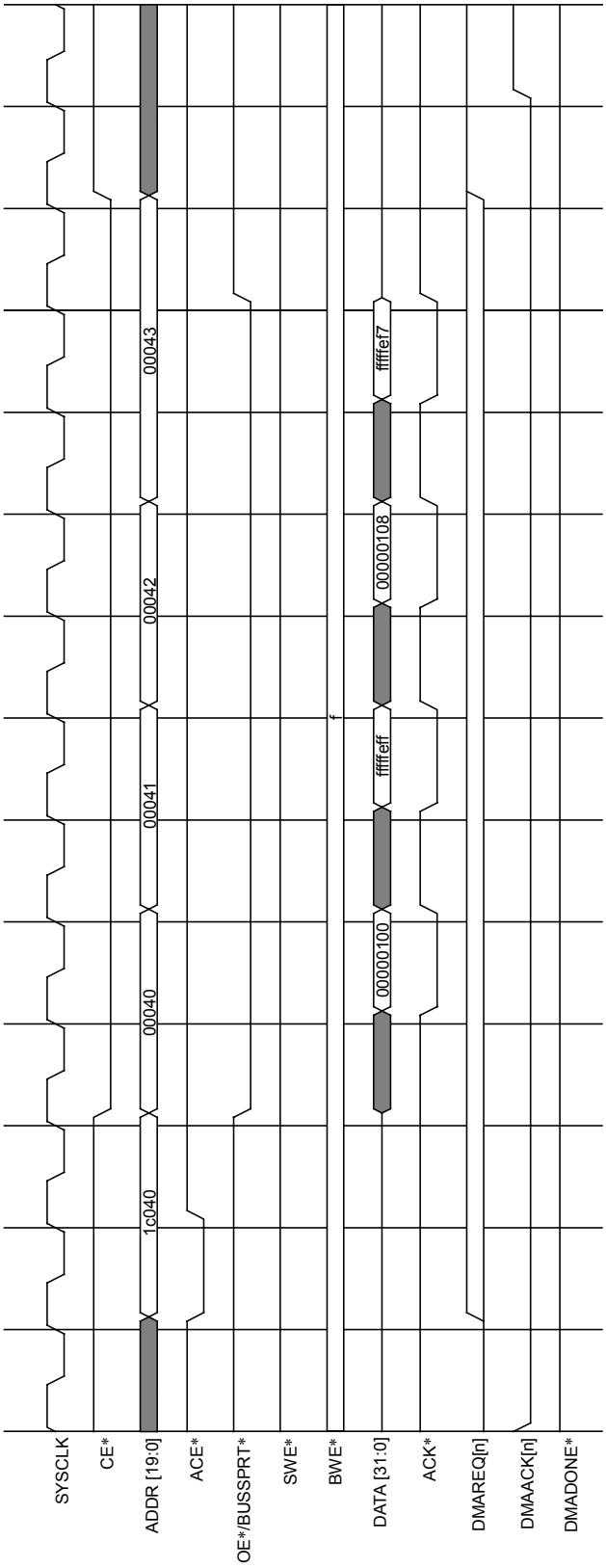


Figure 8.5.4 Single Address Burst Transfer from Memory to I/O
(Burst Read of 4-word Data from 32-bit ROM)

8.5.5 Single Address Burst Transfer from I/O to Memory (32-bit SRAM)

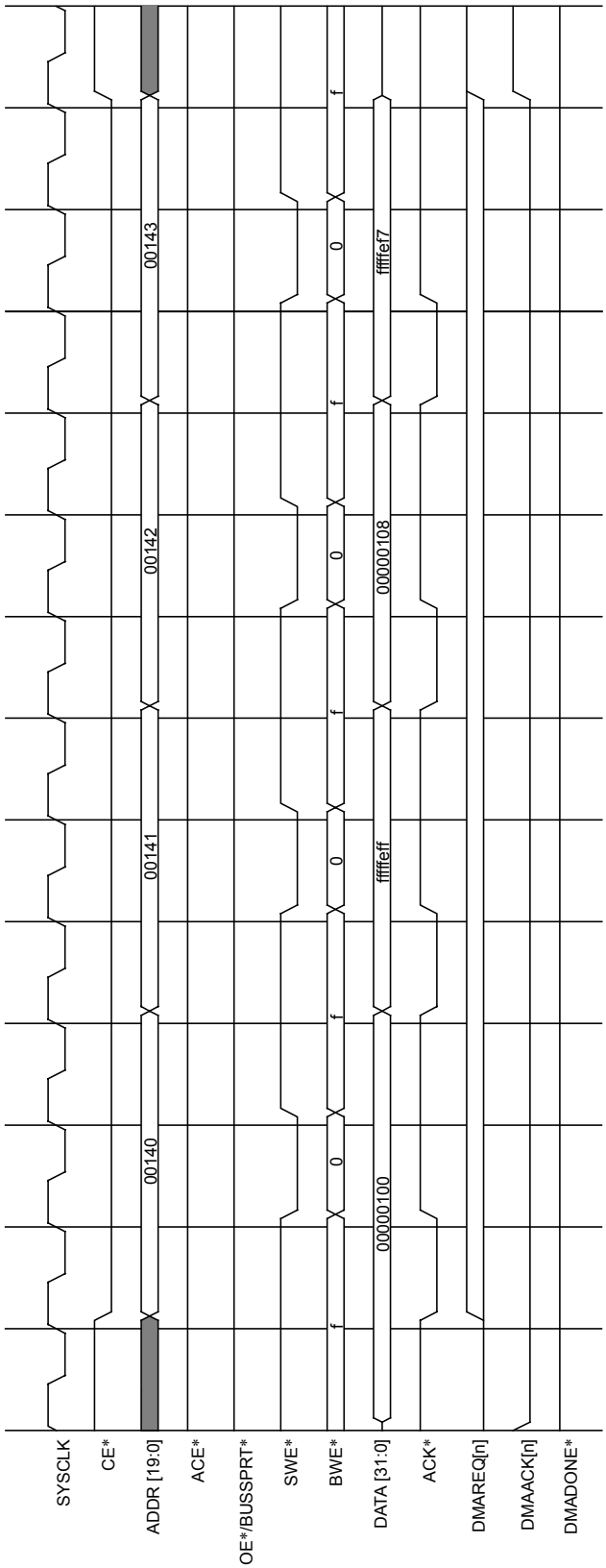


Figure 8.5.5 Single Address Burst Transfer from I/O to Memory
(Burst Write of 4-word Data from 32-bit SRAM)

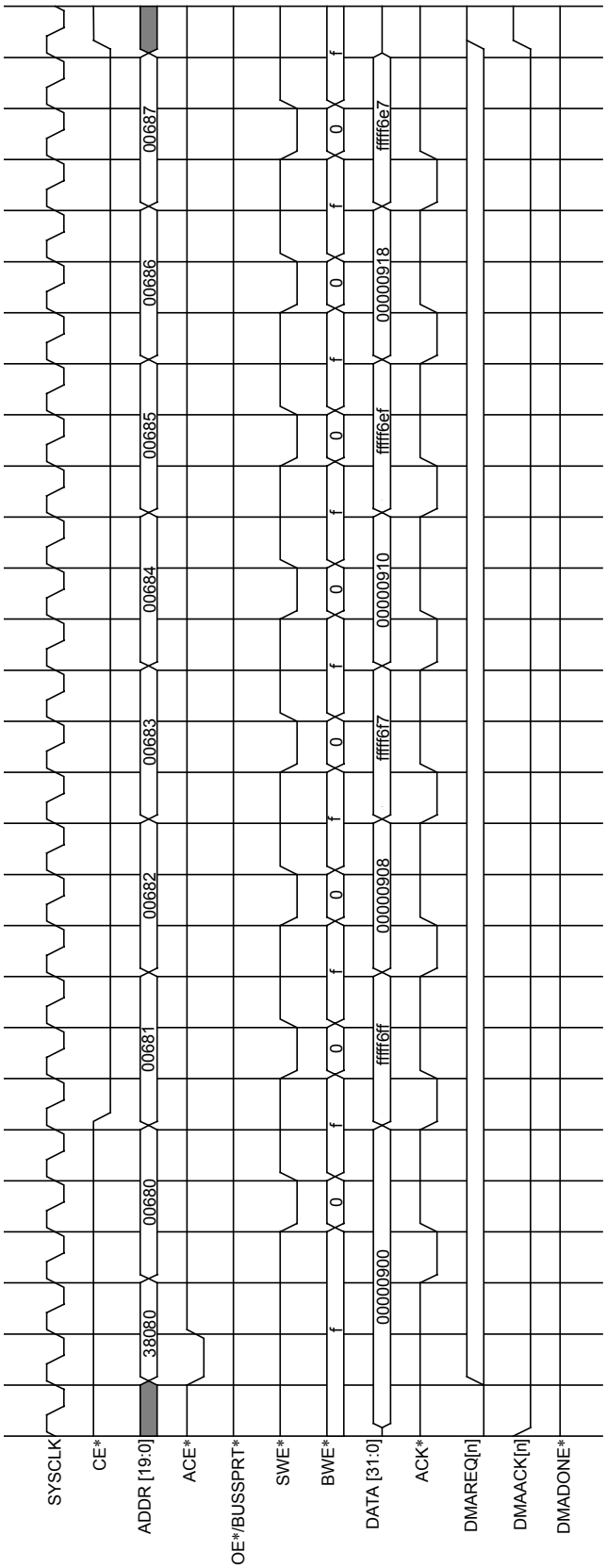


Figure 8.5.6 Single Address Burst Transfer from I/O to Memory
(Burst Write of 8-word Data to 32-bit SRAM)

8.5.6 Single Address Single Transfer from Memory to I/O (16-bit ROM)

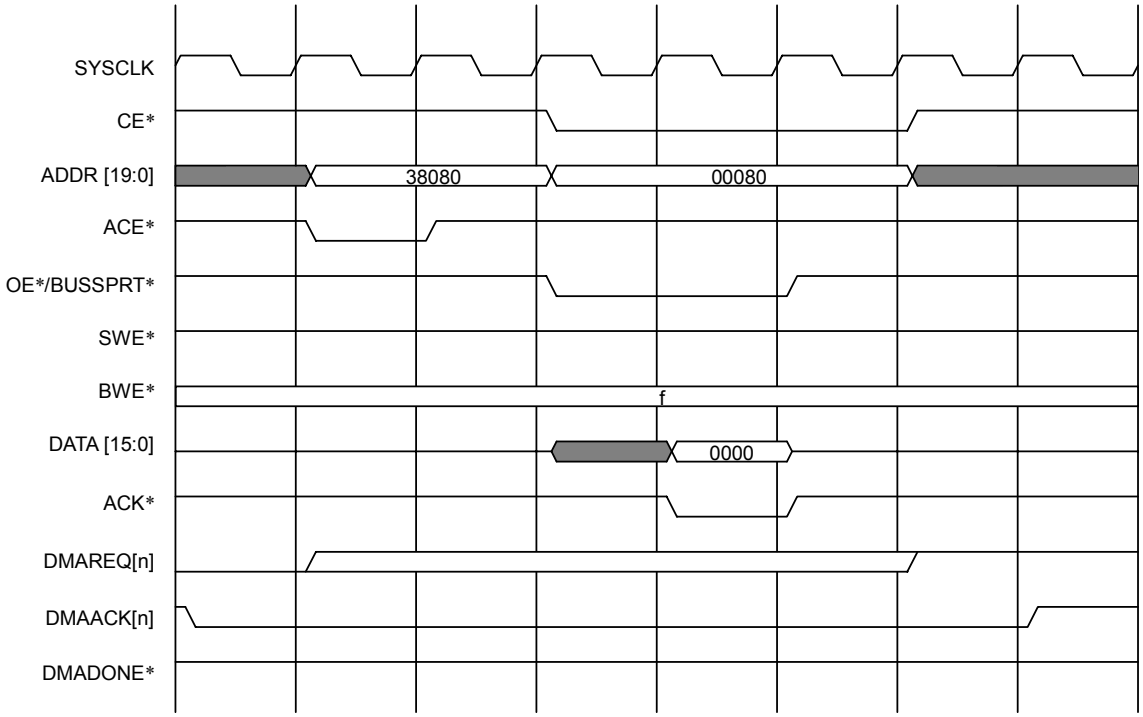


Figure 8.5.7 Single Address Single Transfer from Memory to I/O
(Single Read from 16-bit ROM to 16-bit Data)

8.5.7 Single Address Single Transfer from I/O to Memory (16-bit SRAM)

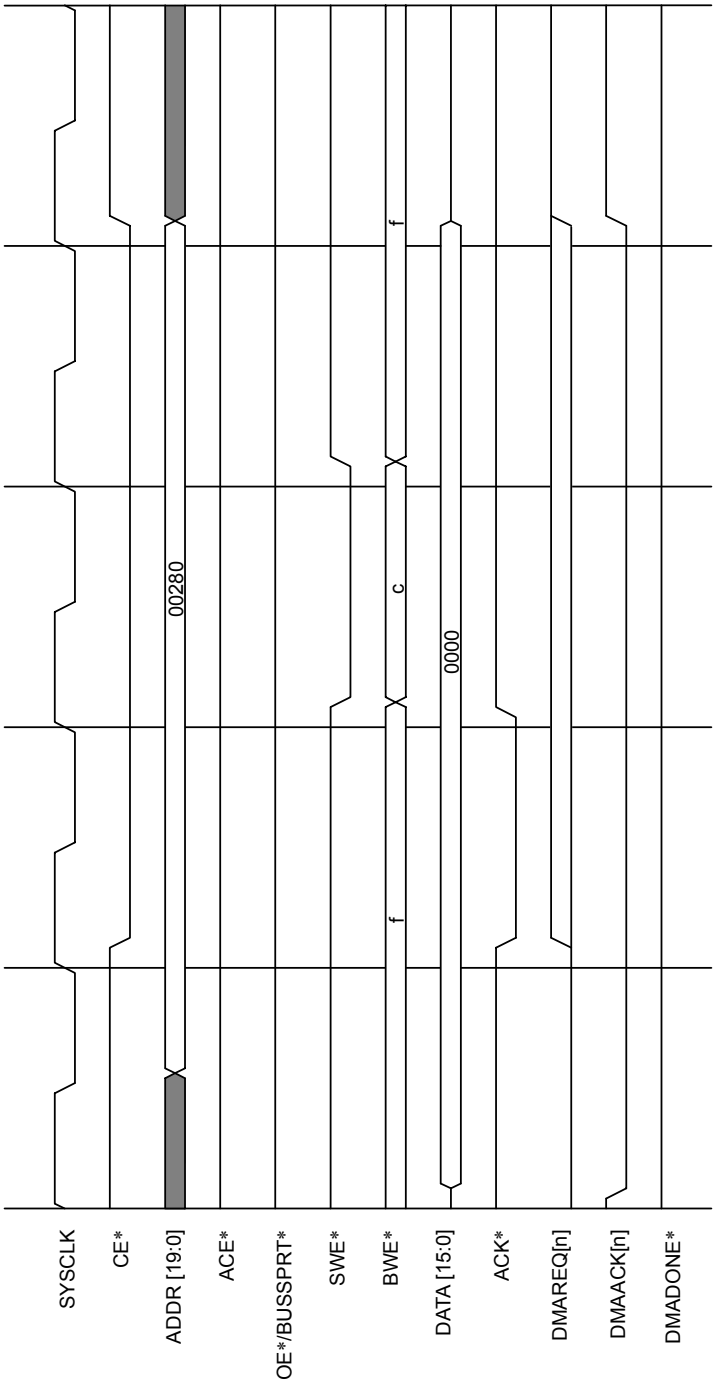


Figure 8.5.8 Single Address Single Transfer from I/O to Memory
(Single Write of 16-bit Data to 16-bit SRAM)

8.5.8 Single Address Single Transfer from Memory to I/O (32-bit Half Speed ROM)

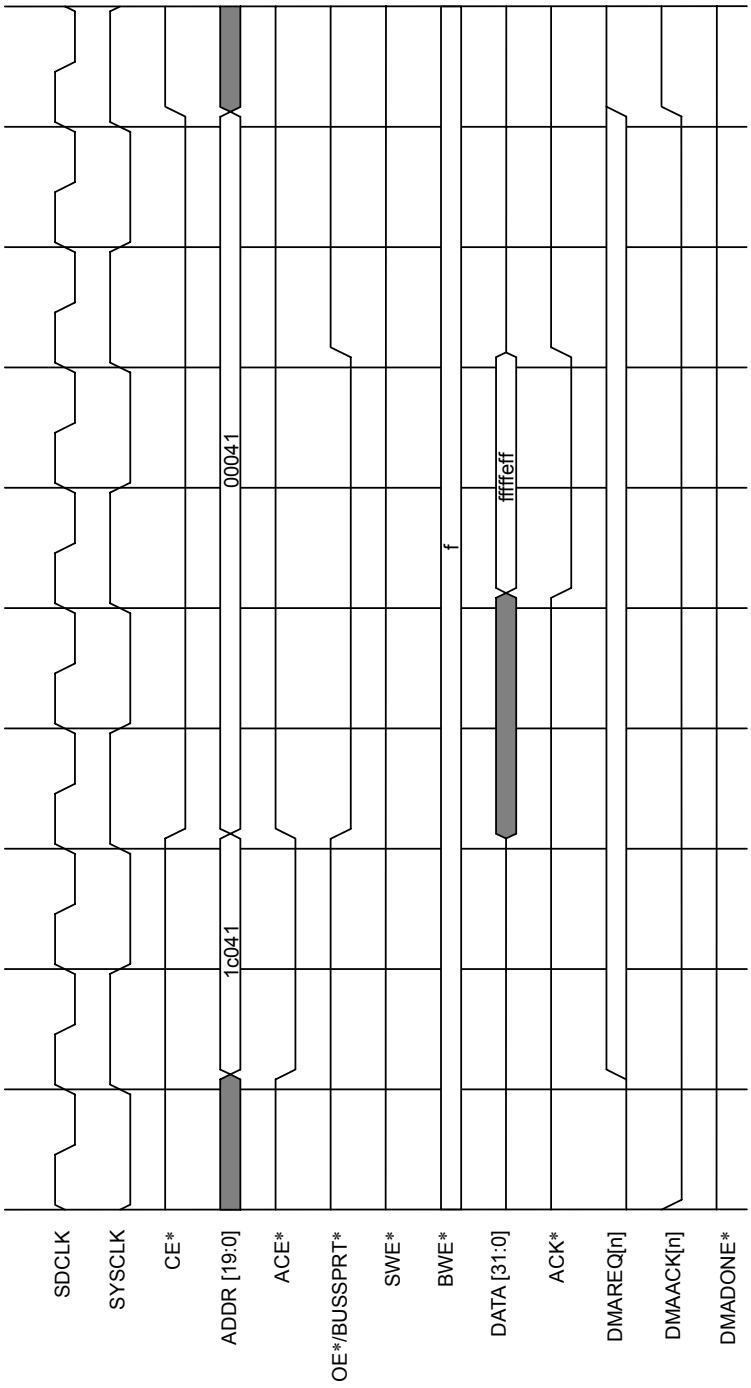


Figure 8.5.9 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 32-bit Half Speed ROM)

8.5.9 Single Address Single Transfer from I/O to Memory (32-bit Half Speed SRAM)

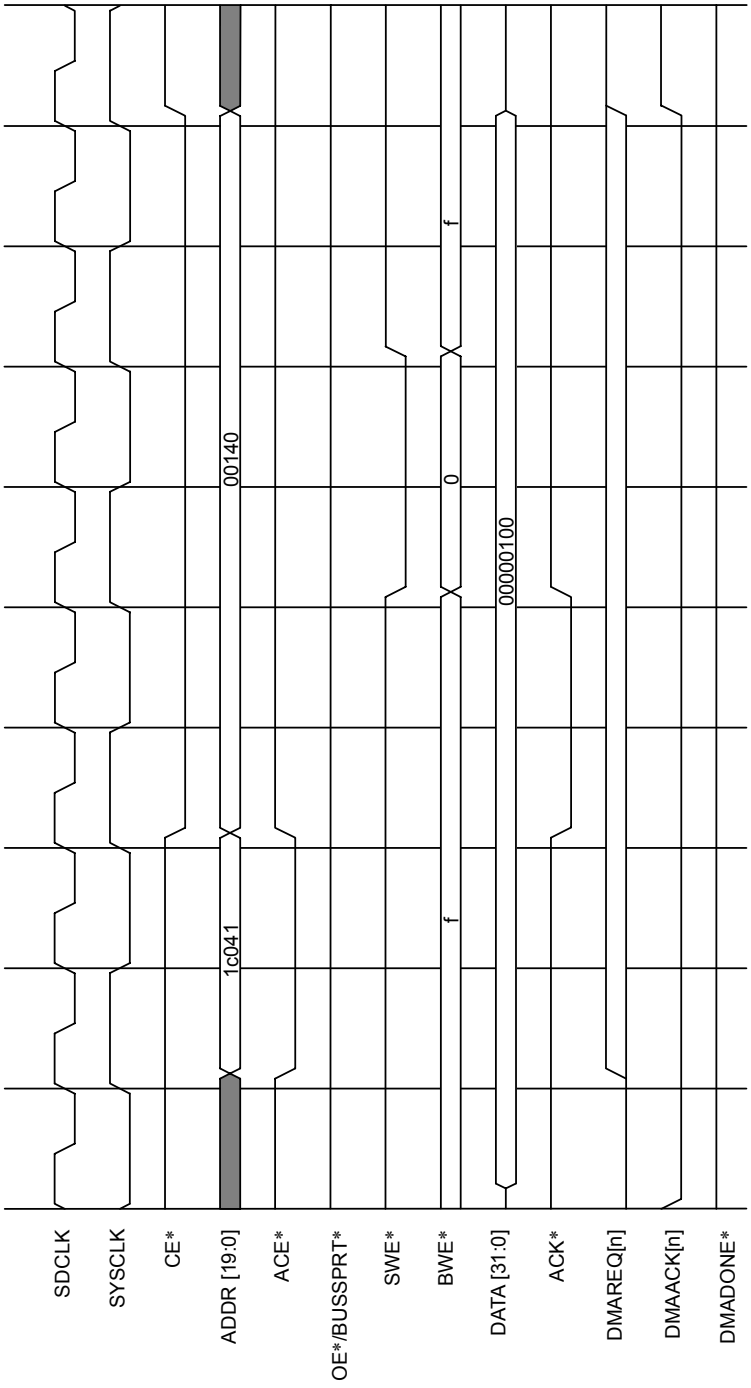


Figure 8.5.10 Single Address Single Transfer from I/O to Memory
(Single Write of 32-bit Data to 32-bit Half Speed SRAM)

8.5.10 Single Address Single Transfer from Memory to I/O (64-bit SRAM)

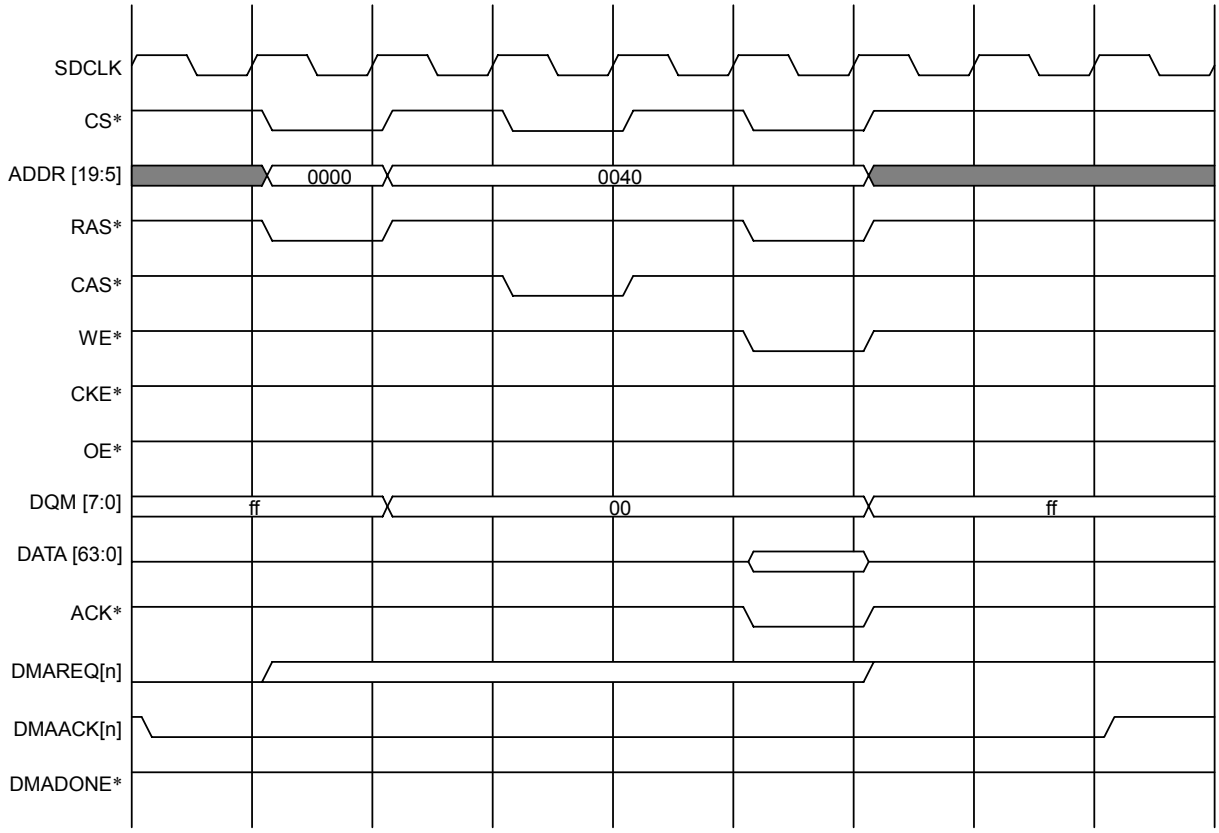


Figure 8.5.11 Single Address Single Transfer from Memory to I/O
(Single Read of 64-bit Data from 64-bit SDRAM)

8.5.11 Single Address Single Transfer from I/O to Memory (64-bit SDRAM)

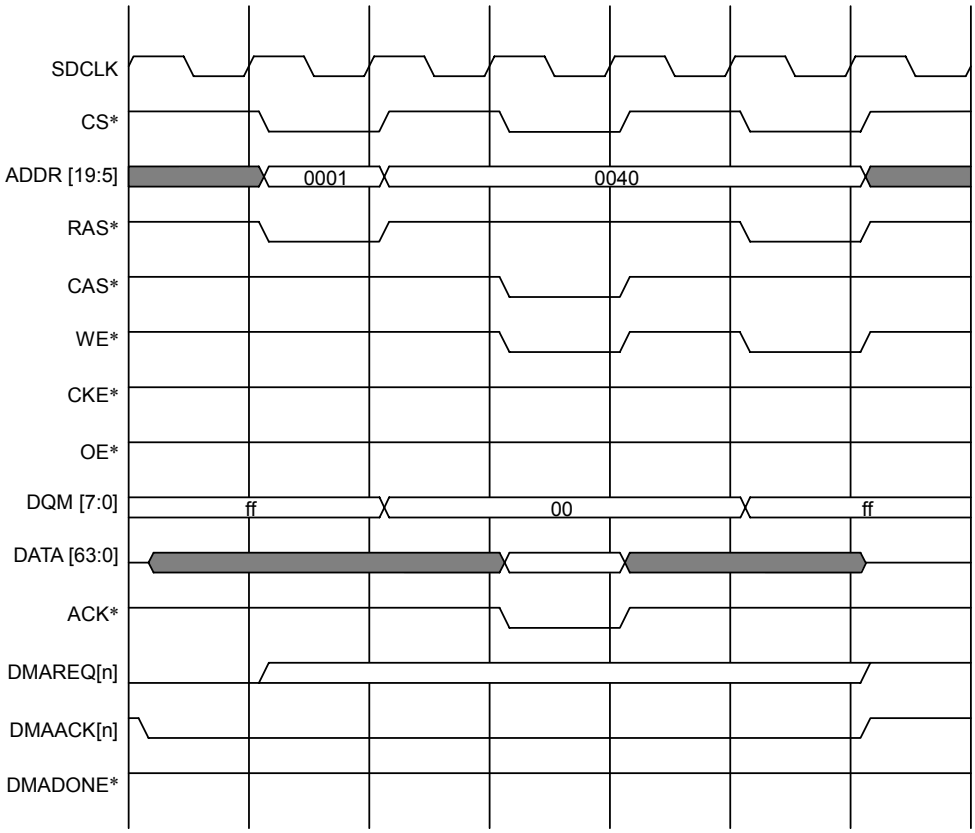


Figure 8.5.12 Single Address Single Transfer from I/O to Memory
(Single Write of 64-bit Data to 64-bit SDRAM)

8.5.12 Single Address Single Transfer from Memory to I/O of Last Cycle when DMADONE* Signal is Set to Output

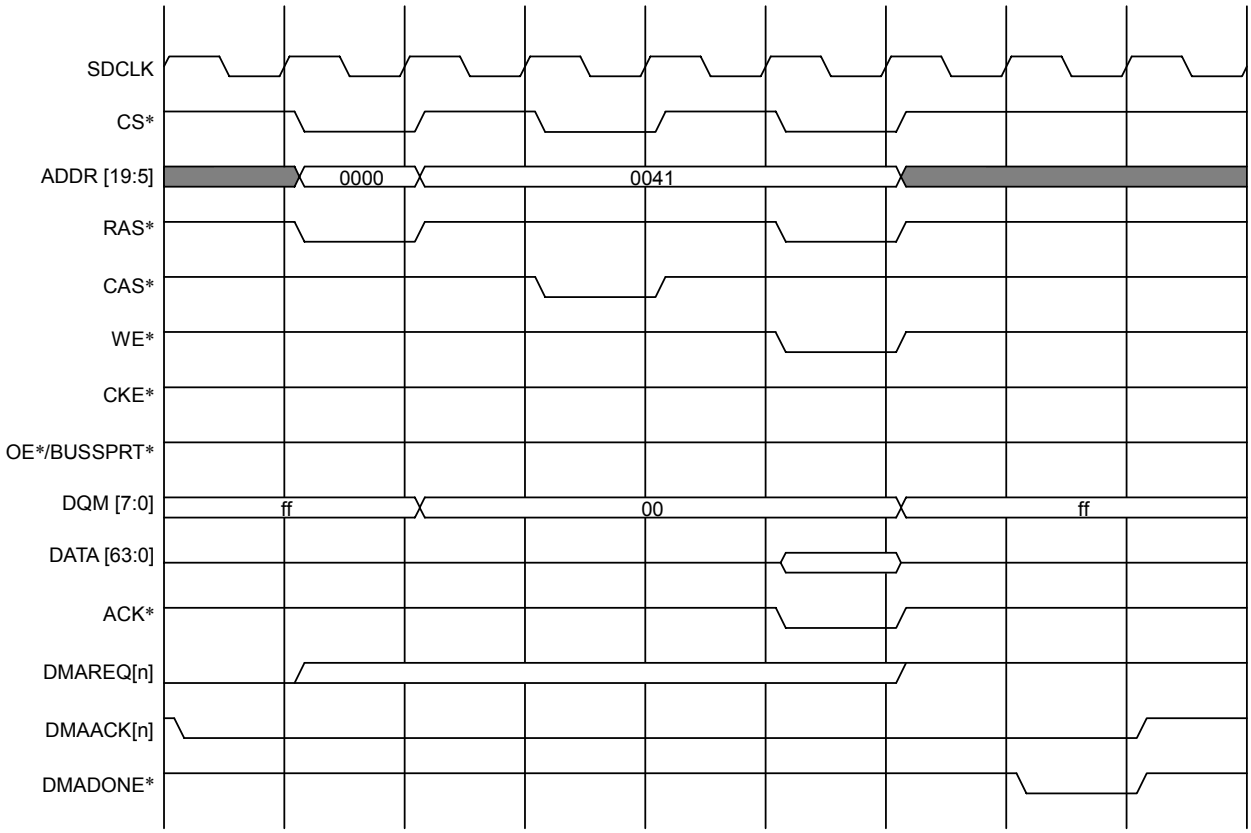


Figure 8.5.13 Single Address Single Transfer from Memory to I/O
(Single Read of 64-bit Data from 64-bit SDRAM)

8.5.13 Single Address Single Transfer from Memory to I/O (32-bit SDRAM)

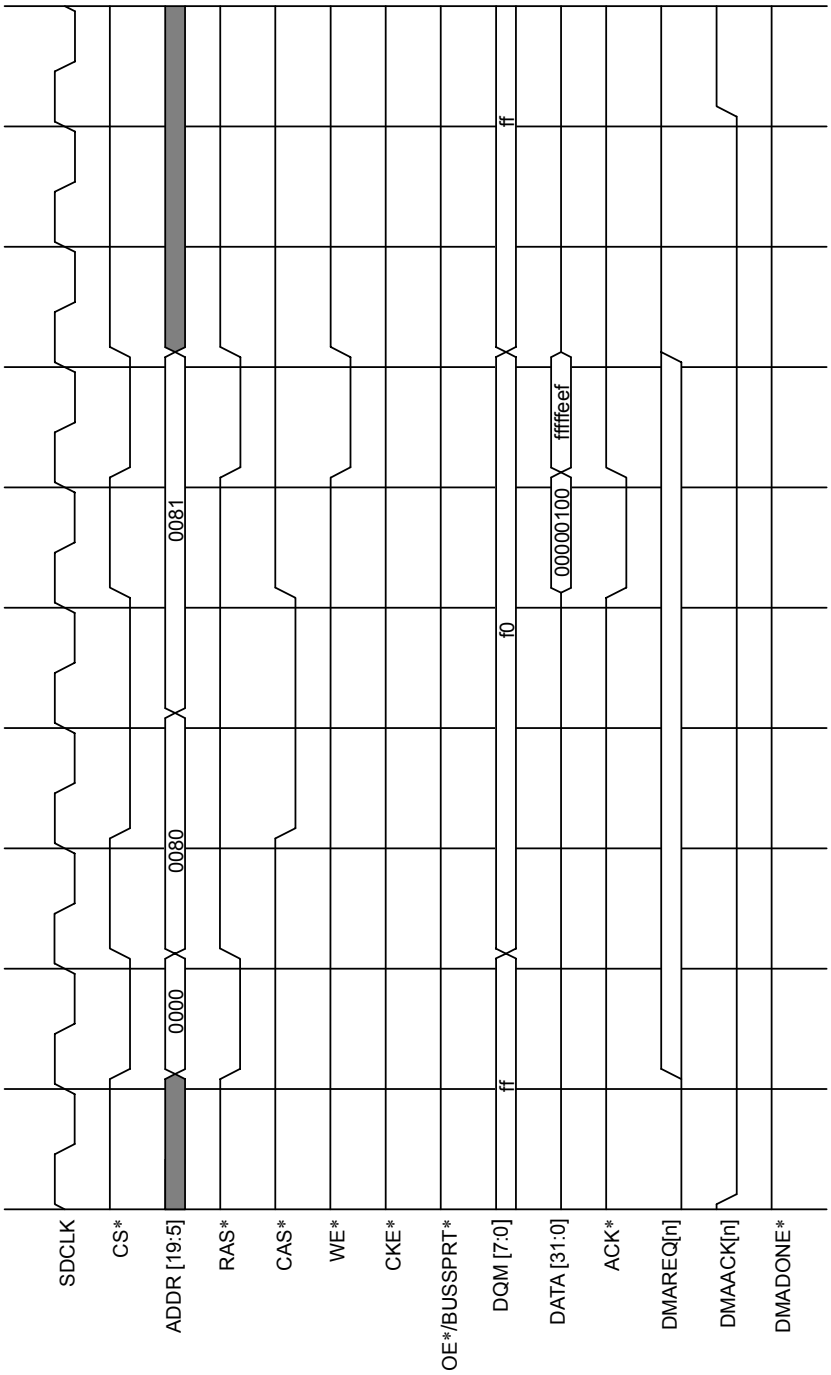


Figure 8.5.14 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 32-bit SDRAM)

8.5.14 Single Address Single Transfer from I/O to Memory (32-bit SDRAM)

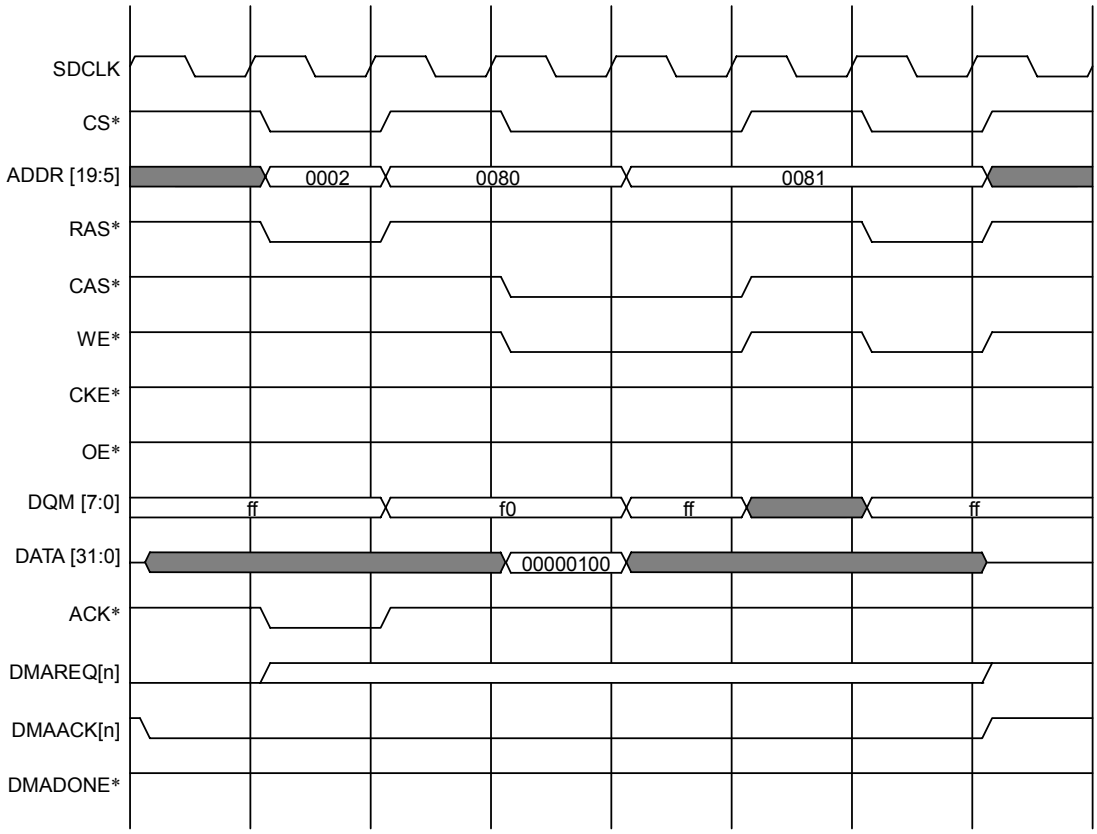


Figure 8.5.15 Single Address Single Transfer from I/O to Memory
(Single Write of 32-bit Data to 32-bit SDRAM)

8.5.15 External I/O Device – SRAM Dual Address Transfer

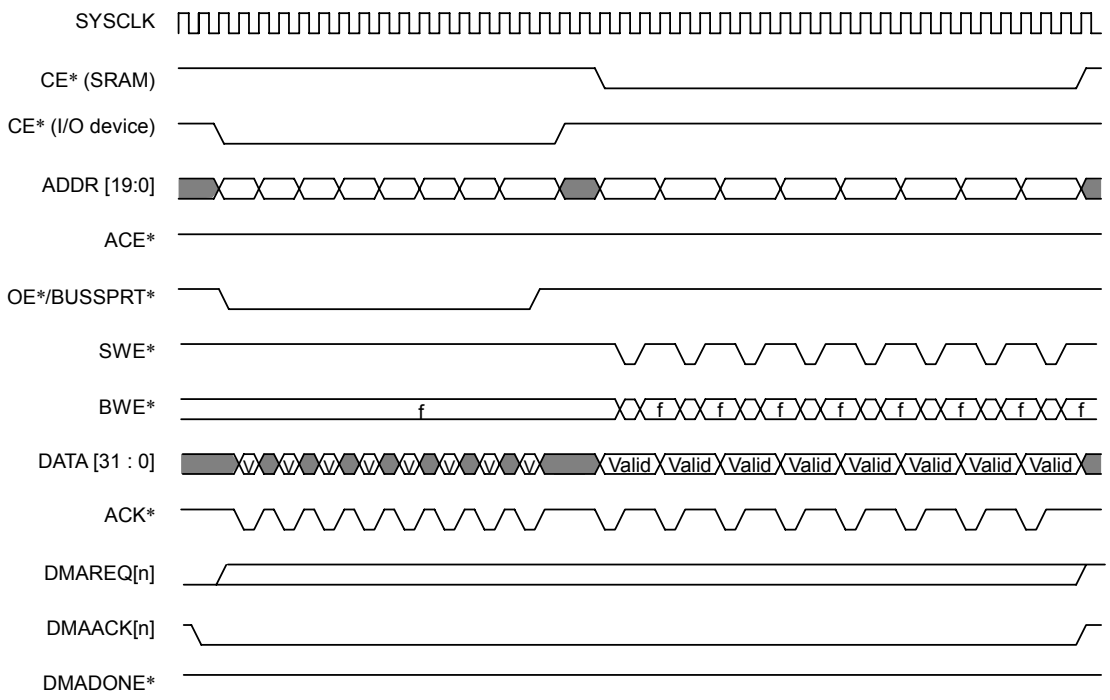


Figure 8.5.16 Dual Address Transfer from External I/O Device to SRAM
(8-word Burst Transfer to 32-bit Bus SRAM)

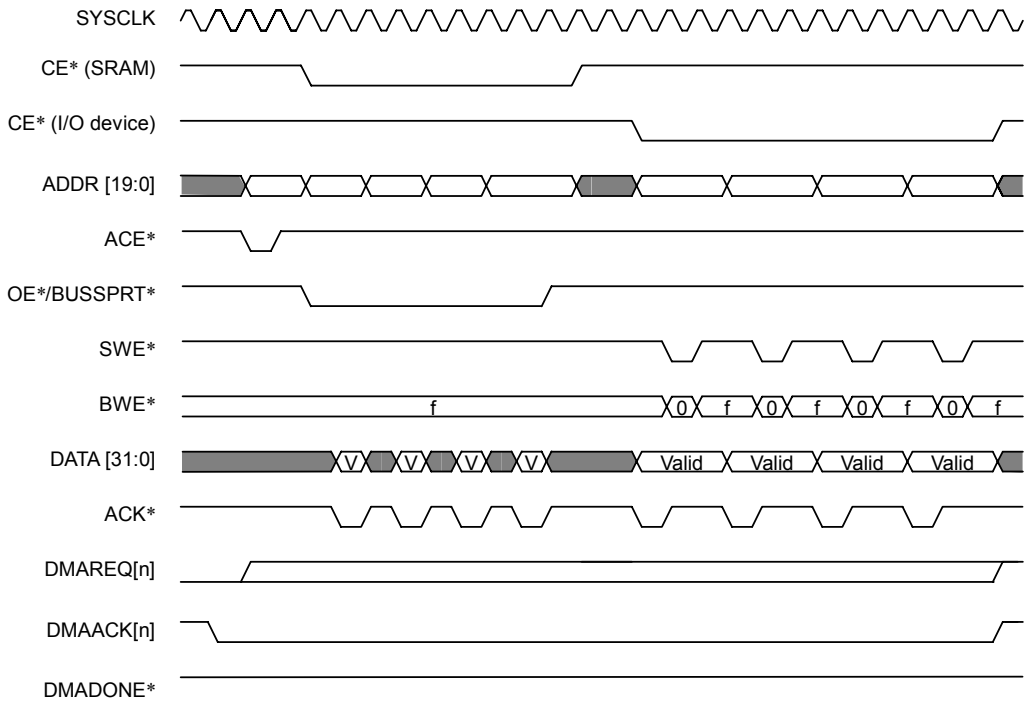


Figure 8.5.17 Dual Address Transfer from SRAM to External I/O Device
(4-word Burst Transfer from 32-bit Bus SRAM)

8.5.16 External I/O Device – SDRAM Dual Address Transfer

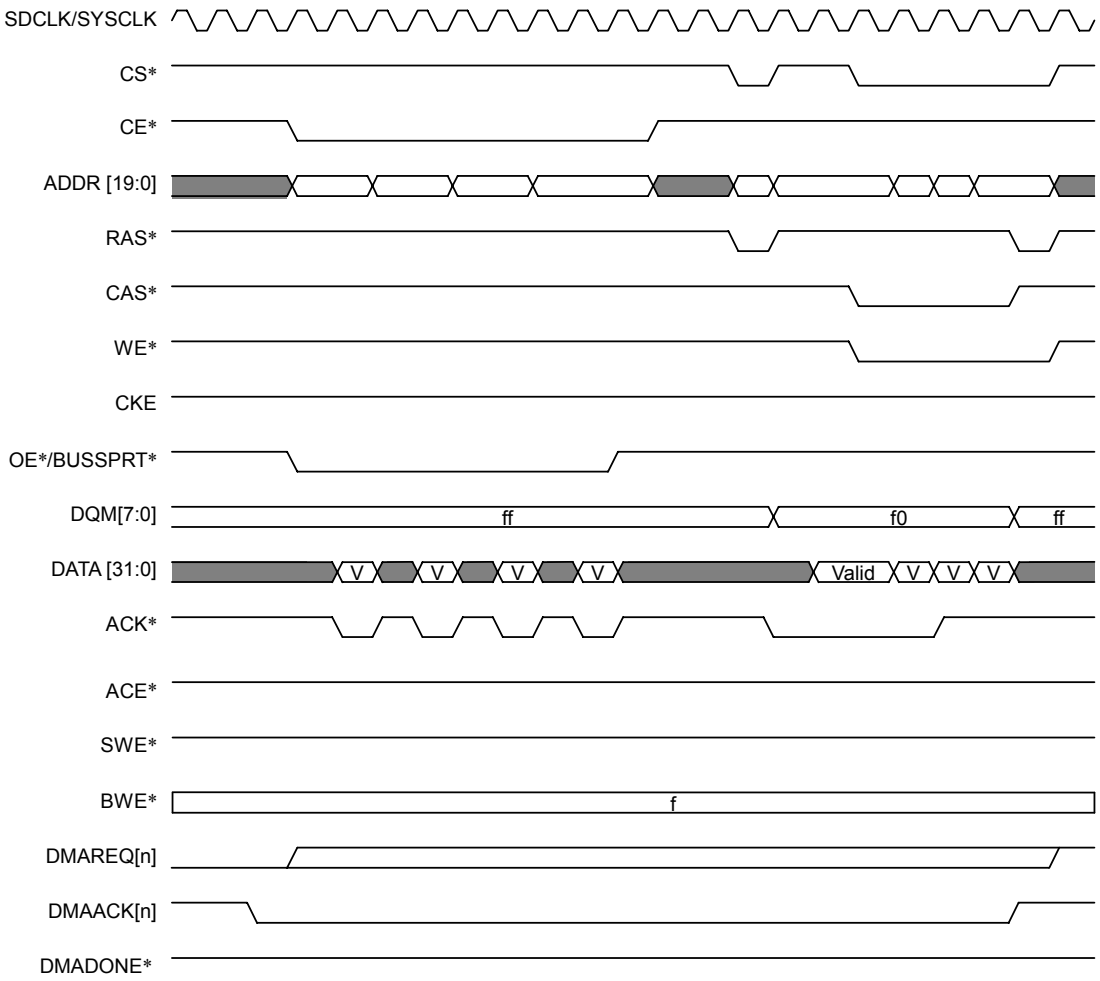


Figure 8.5.18 Dual Address Transfer from External I/O Device to SDRAM
(4-word Burst Transfer to 32-bit SDRAM)

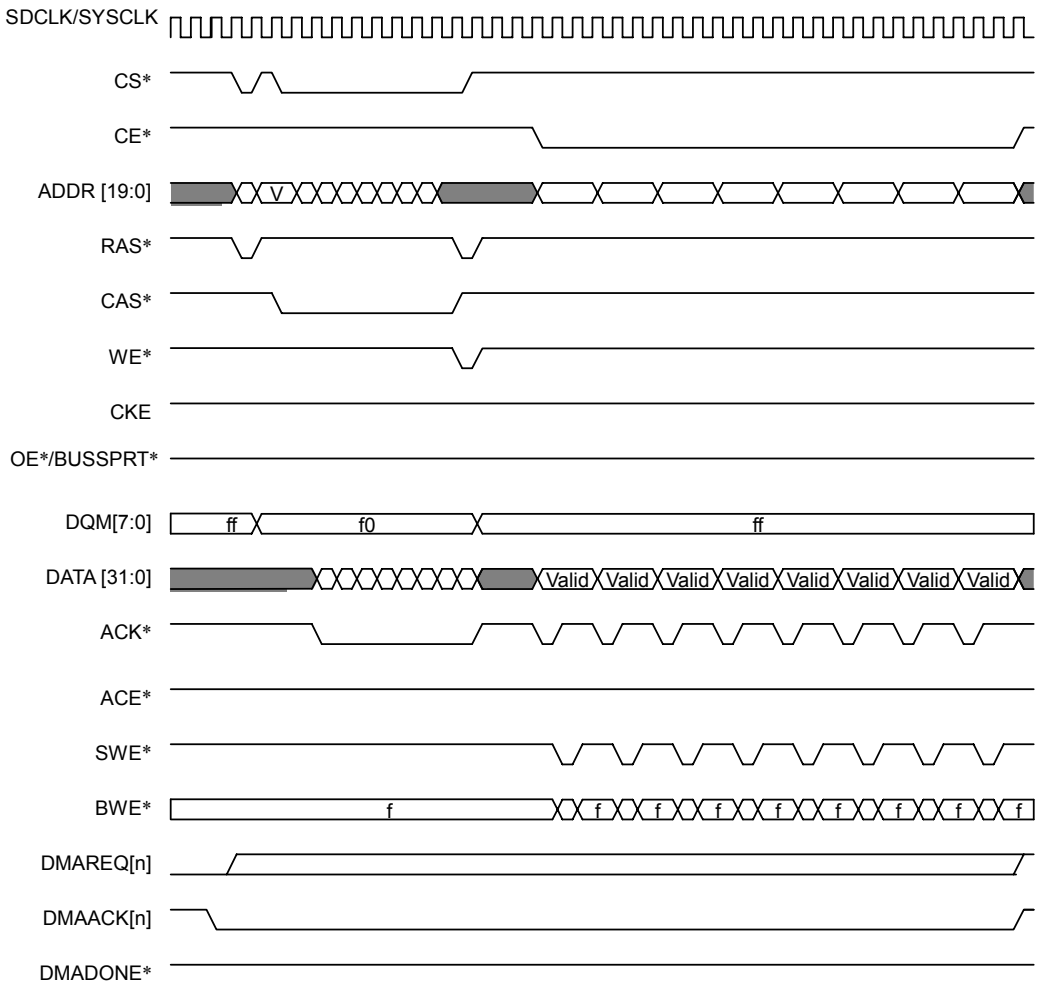


Figure 8.5.19 Dual Address Transfer from SDRAM to External I/O Device
(8-word Burst Transfer from 32-bit SDRAM)

8.5.17 External I/O Device (Non-burst) – SDRAM Dual Address Transfer

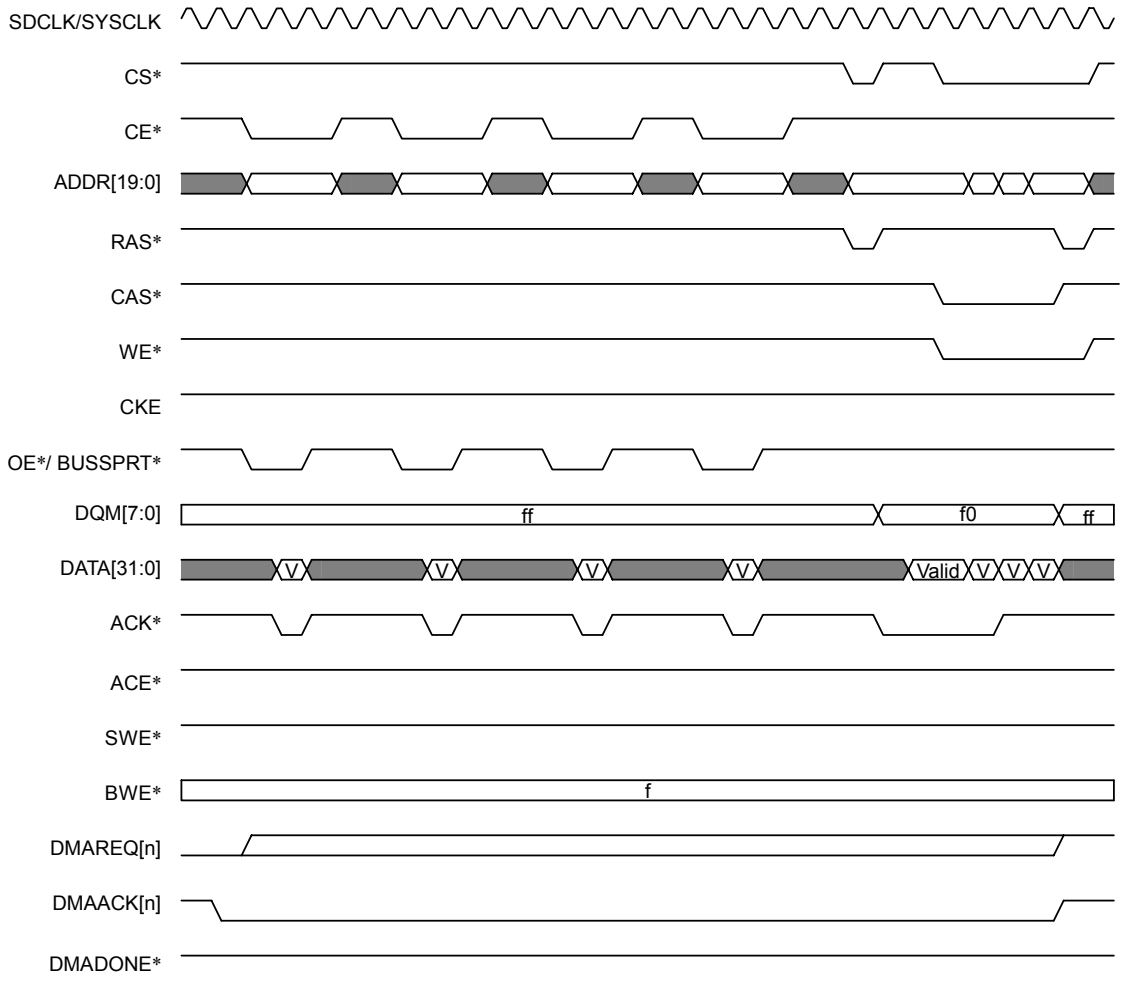


Figure 8.5.20 Dual Address Transfer from External I/O Device (Non-Burst) to SDRAM
(4-word Burst Transfer to 32-bit SDRAM: Set DMCCRn.SBINH to “1”)

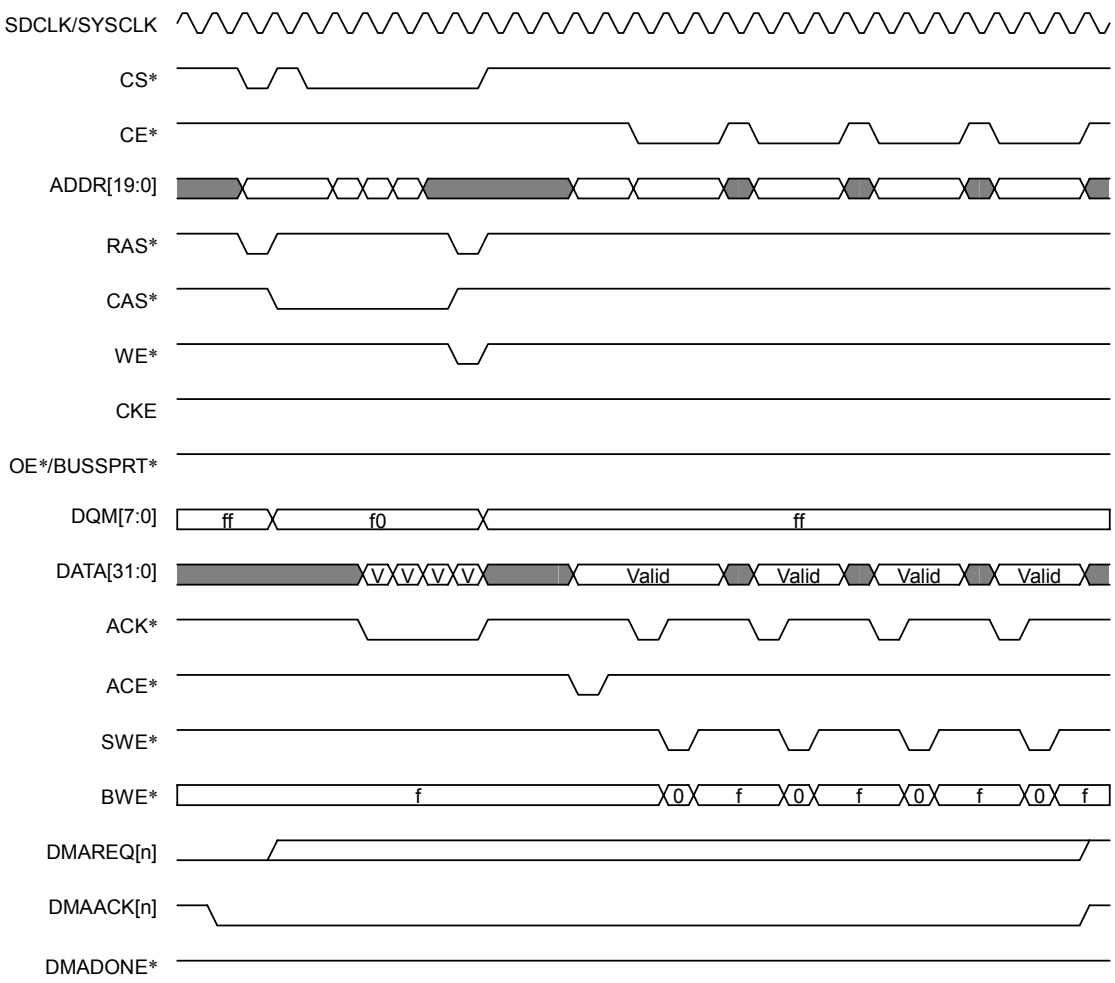


Figure 8.5.21 Dual Address Transfer from SDRAM to External I/O Device
(4-word Burst Transfer from 32-bit SDRAM: Set DMCCRn.DBINH to “1”)

9. SDRAM Controller

9.1 Characteristics

The SDRAM Controller (SDRAMC) generates the control signals required to interface with the SDRAM. There are a total of four channels, which can each be operated independently. The SDRAM Controller supports various bus configurations and a memory size of up to 2 GB.

The SDRAM has the following characteristics.

- Memory clock (SDCLK) frequency: 50 – 133 MHz (For relationship between CPU clock and memory clock, see Section 6.1)
- Four independent memory channels
- Can use registered DIMM
- Selectable data bus width for each channel: 64-bit/32-bit
- Supports critical word first access of the TX49/H3 core
- Supports DMAC special Burst access (address decrement/fix)
- Programmable SDRAM timing latency
  Can set timing to match the clock frequency used and the memory speed. Can realize a system with optimized memory performance.
- Can write to any byte during Single or Burst Write operation. This feature is controlled by the DQM signal.
- Can set the refresh cycle to be programmable.
- SDRAM refresh mode: both auto refresh and self refresh are possible.
- Low power consumption mode: can select between self refresh or pre-charge power down
- SDRAM Burst length: fixed to "2"
- SDRAM addressing mode: Fixed to the Sequential mode
- Supports systems with high fan-out
  Supports two selectable data read-back buses and supports the Slow Write Burst Mode in order to handle data buses with large load. In order to maintain timing consistency during Read operation, it is possible to select whether to use the feedback clock to latch data or to by-pass this latch path. Two clock cycles are used for each Write operation when in the Slow Write Burst Mode.
- Can use the ECC or parity generation/check functions.
- Can select EC (Error Check only), ECC (Error Check and Correct), or ECC + Scrub (write correction data back to memory) when using the ECC function.
- Can select Odd parity/Even parity when using the Parity function.

9.2 Block Diagram

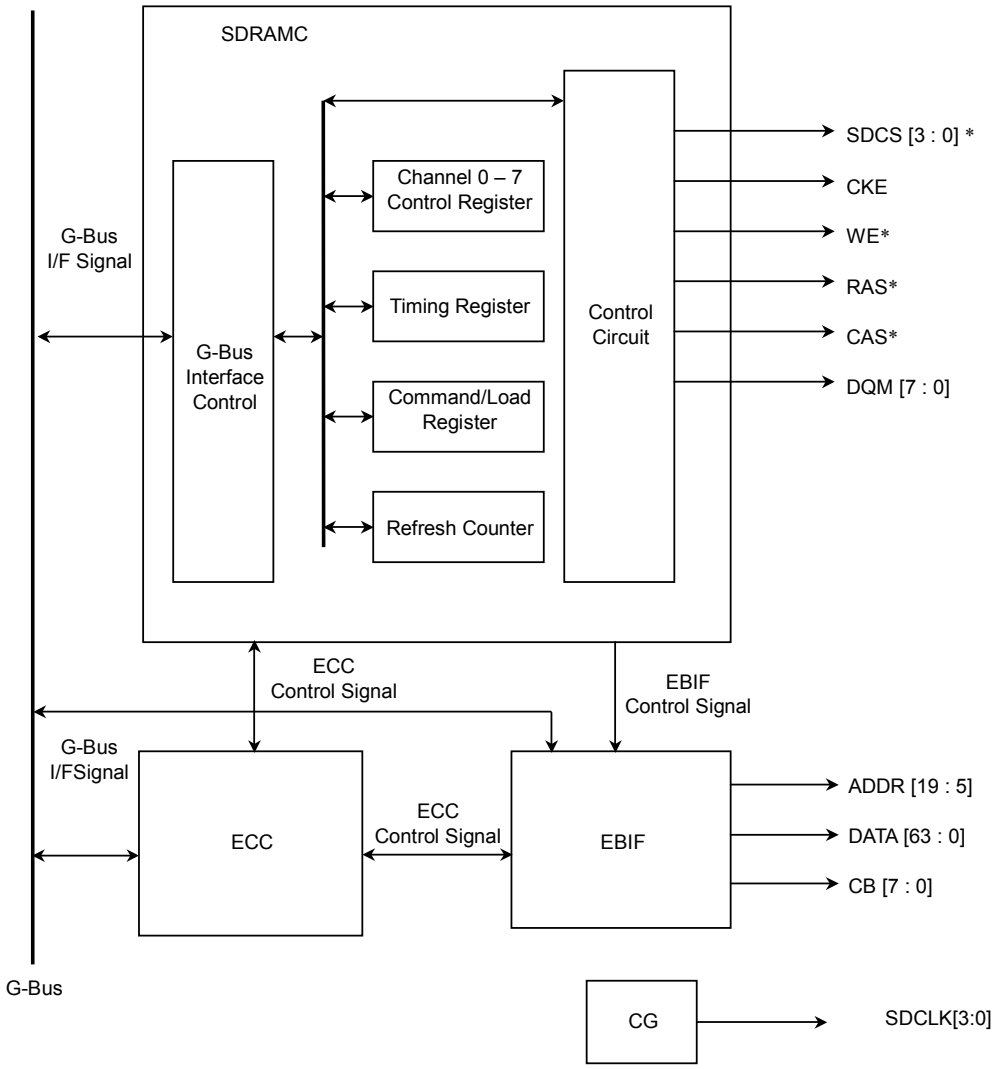


Figure 9.2.1 Block Diagram of SDRAMC

9.3 Detailed Explanation

9.3.1 Supported SDRAM configurations

This controller supports the SDRAM configurations listed below in Table 9.3.1.

The MW field of the SDRAM Channel Control Register (SDCCRn) can be used to separately set the data bus width for each channel to either 64 bits or 32 bits.

DATA[31:0] and DQM[3:0] are used when using a 32-bit data bus. DQM[7:4] output High. DATA[63:32] output an undefined value when DATA[31:0] become the output, but enter the High-Z state when DATA[31:0] are the input. When in the Big Endian Mode, first external access of the upper word (bits 63:32) of the internal data bus is performed, then external access of the lower word (bits 31:0) is performed. When in the Little Endian Mode, first external access of the lower word (bits 31:0) is performed, then external access of the upper word (bits 63:32) is performed. When using a 32-bit data bus, two external access will always be performed even when accessing less than 32 bits of data.

The maximum memory capacity per channel when a 64-bit data bus is configured is 1 GBytes when using 16 512-Mbit SDRAMs with a 4-bit data bus. The total maximum memory capacity is 4 GBytes when totaling up the four channels.

Table 9.3.1 Supported SDRAM Configurations

SDRAM Configuration			Row Address (bit)	Column Address (bit)	Remarks
16 Mbit	2-bank	1 M × 16	11	8	
		2 M × 8	11	9	
		4 M × 4	11	10	See Note 1
64 Mbit	2-bank	2 M × 32	11	9	
		2 M × 32	12	8	
		4 M × 16	11	10	
		4 M × 16	13	8	
		8 M × 8	13	9	
		16 M × 4	13	10	See Note 1
	4-bank	2 M × 32	11	8	
		4 M × 16	12	8	
		8 M × 8	12	9	
		16 M × 4	12	10	See Note 1
128 Mbit	4-bank	4 M × 32	12	8	
		8 M × 16	12	9	
		16 M × 8	12	10	
		32 M × 4	12	11	See Note 1
256 Mbit	4-bank	16 M × 16	13	9	
		32 M × 8	13	10	
		64 M × 4	13	11	See Note 1,2
512 Mbit	4-bank	32 M × 16	13	10	
		64 M × 8	13	11	See Note 2
		128 M × 4	13	12	See Note 1,2

Note1: The SDRAM Controller logic-wise does support these configurations, but please design carefully since the memory bus load will be large.

Note2: This SDRAM configuration has 512 Mbytes of memory on a channel. If it is mapped to physical address space beginning with address 0, it overlaps the address space for the ROM wherein the bootstrap vectors reside.

9.3.2 Address Mapping

9.3.2.1 Physical Address Mapping

It is possible to map each of the four channels to an arbitrary physical address using the Base Address field (SDCCRn.BA[35:21]) of the SDRAM Channel Control Register and the Address Mask Field (SDCCRn.AM[35:21]).

The channel that becomes True in the following equation is selected.

paddr[35:21] & !AM[35:21] = BA[35:21] & !AM[35:21]

In the above equation, "paddr" represents the accessed physical address, "&" represents the AND of each bit, and "!" represents the logical NOT of each bit.

Operation is undefined when multiple channels are simultaneously selected, or when external bus controllers or PCI controllers are simultaneously selected.

9.3.2.2 Address Signal Mapping (64-bit Data Bus)

Table 9.3.2 shows the address signal mapping when using a 64-bit data bus. B0 is used in the bank selection in memory with a two-bank configuration. [B1:B0] are used in the bank selection in memory with a four-bank configuration. Bits with the description "L/H" output High when performing auto-precharging, or output Low when not performing auto-precharging.

Table 9.3.2 Address Signal Mapping (64-bit Data Bus) (1/2)

Row address width = 11 Column address width = 8															
Address bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	23	21	20	L/H	L/H	L/H	10	9	8	7	6	5	4	3
Row Address	22	23	21	20	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 11 Column Address Width = 9															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	23	23	23	L/H	23	22	10	9	8	7	6	5	4	3
Row Address	23	23	23	23	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 11 Column Address Width = 10															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	23	23	24	L/H	23	22	10	9	8	7	6	5	4	3
Row Address	24	23	23	24	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 12 Column Address Width = 8															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	24	23	22	L/H	24	23	10	9	8	7	6	5	4	3
Row Address	23	24	23	22	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 12 Column Address Width = 9															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	24	22	L/H	24	23	10	9	8	7	6	5	4	3
Row Address	24	25	24	22	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 12 Column Address Width = 10															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	25	26	25	22	L/H	24	23	10	9	8	7	6	5	4	3
Row Address	25	26	25	22	21	20	19	18	17	16	15	14	13	12	11

Table 9.3.2 Address Signal Mapping (64-bit Data Bus) (2/2)

Row Address Width = 12 Column Address Width = 11															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	26	27	26	25	L/H	24	23	10	9	8	7	6	5	4	3
Row Address	26	27	26	22	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 13 Column Address Width = 8															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	23	22	L/H	25	24	10	9	8	7	6	5	4	3
Row Address	24	25	23	22	21	20	19	18	17	16	15	14	13	12	11

Row Address = 13 Column Address Width = 9															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	25	26	23	22	L/H	25	24	10	9	8	7	6	5	4	3
Row Address	25	26	23	22	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 13 Column Address Width = 10															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	26	27	23	22	L/H	25	24	10	9	8	7	6	5	4	3
Row Address	26	27	23	22	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 13 Column Address Width = 11															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	27	28	23	26	L/H	25	24	10	9	8	7	6	5	4	3
Row Address	27	28	23	22	21	20	19	18	17	16	15	14	13	12	11

Row Address Width = 13 Column Address Width = 12															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	28	29	27	26	L/H	25	24	10	9	8	7	6	5	4	3
Row Address	28	29	23	22	21	20	19	18	17	16	15	14	13	12	11

9.3.2.3 Address Signal Mapping (32-bit Data Bus)

Table 9.3.3 shows the address signal mapping when using a 32-bit data bus. B0 is used in the bank selection in memory with a two-bank configuration. [B1:B0] are used in the bank selection in memory with a four-bank configuration. Bits with the description "L/H" output High when performing auto-precharging, or output Low when not performing auto-precharging.

Table 9.3.3 Address Signal Mapping (32-bit Data Bus) (1/2)

Row Address Width = 11 Column Address Width = 8															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	21	22	20	19	L/H	L/H	L/H	9	8	7	6	5	4	3	2
Row Address	21	22	20	19	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 11 Column Address Width = 9															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	22	22	22	L/H	22	21	9	8	7	6	5	4	3	2
Row Address	22	22	22	22	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 11 Column Address Width = 10															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	22	22	23	L/H	22	21	9	8	7	6	5	4	3	2
Row Address	23	22	22	23	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 12 Column Address Width = 8															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	23	22	21	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	22	23	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 12 Column Address Width = 9															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	24	23	21	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	23	24	23	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 12 Column Address Width = 10															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	24	21	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	24	25	24	21	20	19	18	17	16	15	14	13	12	11	10

Table 9.3.3 Address Signal Mapping (32-bit Data Bus) (2/2)

Row Address Width = 12 Column Address Width = 11															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	25	26	25	24	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	25	26	25	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13 Column Address Width = 8															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	24	22	21	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	23	24	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13 Column Address Width = 9															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	22	21	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	24	25	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13 Column Address Width = 10															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	25	26	22	21	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	25	26	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13 Column Address Width = 11															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	26	27	22	25	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	26	27	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13 Column Address Width = 12															
Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	15 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	29	28	26	25	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	29	28	22	21	20	19	18	17	16	15	14	13	12	11	10

9.3.3 Initialization of SDRAM

The TX4938 Command Register has functions for generating the cycles required for initializing SDRAM. Using software to set each register makes it possible to execute initial settings at a particular timing.

1 Set the SDRAM Channel Control Register (SDCCRn).

2 Set the SDRAM Timing Register (SDCTR). This timing setting is applied to all channels, so please set it to the slowest memory device.

3 Use the SDRAM Command Register (SDCCMD) to issue the Pre-charge All command.

4 Issue the Set Mode Register command in the same manner.

5 Set the refresh count required to initialize SDRAM to the refresh counter (SDCTR.RC)[1] and set the refresh cycle (SDCTR.RP).[2] [3]

6 Wait until the refresh counter returns to "0."

7 Set the refresh cycle (SDCTR.RP) to the proper value.

[1] The number of refresh operations can be counted using the refresh counter. With this function, it is no longer necessary to assemble special timing groups in the software when counting refresh operations.

[2] Setting the refresh cycle to a small value makes it possible to expedite completion of the refresh cycle required for SDRAM initialization. As described above, please set normal values after the required number of refresh cycles have been generated.

[3] Refresh requests have priority over all other SDRAM Controller access requests. Please do not set the memory refresh cycle to an unnecessarily short value.

9.3.4 Initialization of Memory Data, ECC/Parity

The SDRAMC has functions for simultaneously performing Memory Writes to multiple memory channels. These functions are effective when quickly initializing data memory or ECC/parity memory.

Channels for which both the Channel Enable bit (SDCCRn.CE) and the Master Enable bit (SDCCRn.ME) of the SDRAM Channel Control Register are set become the Master channel. Also, channels for which both the Channel Enable bit (SDCCRn.CE) and the Slave Enable bit (SDCCRn.SE) are set become the Slave channel. See Table 9.3.4 Master/Slave Channel Settings for information regarding the Master/Slave channel settings.

The slave channel is simultaneously written to when the Master channel is written to. Settings of the Master channel are used when in the ECC/Parity mode. Please set to the same value the SDRAM settings of all channels that are simultaneously written to.

Using the DMA Controller and performing 32 double word Burst access is the most efficient way to access the Master channel. The DMAC has registers for setting memory initialization data. When the DMAC is launched by an internal request when in the Single address IO→Memory Transfer mode, the data set in this register are written to memory. See Chapter 8 "DMA Controller" for more information.

Table 9.3.4 Master/Slave Channel Settings

SDCCRn.CE (Channel Enable)	SDCCRn.ME (Master Enable)	SDCCRn.SE (Slave Enable)	Description
0	X	X	Channel is disabled.
1	0	0	Normal operation is performed.
1	0	1	When a Write operation is executed by a channel where SDCCRn.ME=1, the Write operation is also executed by this channel. Normal operation is performed when this channel is Active.
1	1	0	When a Write operation is executed by this channel, the Write operation is simultaneously executed by all channels where SDCCRn.SE=1.
1	1	1	When a Write operation is executed by this channel, the Write operation is simultaneously executed by all channels where SDCCRn.SE=1. A Write operation is also simultaneously executed by this channel when the Write operation is executed by another channel where SDCCRn.ME=1.

9.3.5 Low Power Consumption Function

9.3.5.1 Power Down Mode, Self-Refresh Mode

SDRAM has two low power consumption modes called the Power Down mode and the Self-Refresh mode. Memory data is lost in the case of the Power Down mode since Memory Refresh is not performed, but the amount of power consumed is reduced the most. Memory data is not lost in the case of the Self-Refresh mode.

SDRAM is set to the Power Down mode by using the SDRAM Command Register (SDCCMD) to issue the Power Down Mode command. Similarly, SDRAM is set to the Self-Refresh mode by issuing the Self-Refresh Mode command. The SDRAMC terminates internal refresh circuit operation after one of these commands has been issued. Issuing the Normal Mode command returns operation to normal.

When the Power Down Auto Entry bit (SDCTR.PDAE) of the SDRAM Timing Register is set, SDRAM is automatically set to the Power Down mode when memory access is not being performed. The SDRAMC internal refresh circuit will continue operating, so there will be no loss of memory data.

If either the Memory Access, Memory Refresh, or Memory command is executed while SDRAM is set to the Power Down mode or the Self-Refresh mode, then the Power Down mode and Self-Refresh mode will automatically terminate, and memory access will be performed.

After returning from a low power consumption mode that was set by either the Power Down Mode command or the Self-Refresh Mode command, the next memory access starts after 10 SDCLK cycles pass. This latency sufficiently follows the stipulated time from Power Down to first access of the SDRAM.

If setting the Power Down Auto Entry bit automatically causes memory access to be requested when set in the Power Down mode, then add 1 SDCLK cycle more of access latency than when not in the Power Down mode.

9.3.5.2 Advanced CKE

Advanced CKE is a function that speeds up the CKE assertion and deassertion timing by 1 clock cycle. This function is set using the Address CKE bit (SDCTR.ACE) of the SDRAM Timing Register.

Advanced CKE assumes that it will be used in a system where SDRAM data is saved even when the power to the TX4938 itself is cut. Since CKE On/Off becomes 1 cycle faster, it is possible to delay CKE by 1 clock cycle using external power consumption control logic. Please set the SDRAM to the Self-Refresh mode before using this function.

When combining advanced CKE functionality with Power Down Auto Entry functionality and memory access is requested while in the Power Down mode, two more SDCLK cycles of latency are added than would be the case when not in the Power Down mode.

9.3.6 Bus Errors

The SDRAMC detects bus errors in the following situations:

- Bus time-out occurs during Read or Write operation to the SDRAMC
- ECC 2-bit fault error or Parity error occurs during SDRAM Read operation

If a bus error occurs when accessing the SDRAMC, then the SDRAMC will immediately assert the current operation. Then, the current SDRAM cycle will end, remaining SDRAMC operations will be aborted, a Pre-charge All command will be issued to SDRAM, then the SDRAMC will return to the Idle state.

9.3.7 Memory Read and Memory Write

The RAS* signal, CAS* signal, WE*, signal, and ADDR[19:5] signal are set up 1 cycle before the SDCS* signal is asserted in the case of the Read command, Write command, Pre-charge command, or Mode Register Set command. The same set up time is observed even for active commands if the Active Command Ready bit (SDCTR.DA) of the SDRAM Timing Register is set. Figure 9.5.1 is a timing diagram of Single Read operation when the SDCTR.DA bit is cleared. Figure 9.5.2 is a timing diagram of Single Read operation when the SDCTR.DA bit is set.

Burst or Single Read operation is terminated by the Pre-charge Active Bank command. Burst or Single Write operation is terminated by the Auto Pre-charge Command.

9.3.8 Slow Write Burst

When the Slow Write Burst bit (SDCTR.SWB) of the SDRAM Timing Register is cleared, the data changes at each cycle during Burst Write operation (Figure 9.5.6). When the Slow Write Burst bit is set, the data will change every other cycle (Figure 9.5.7).

When the Slow Write Burst bit is set, all Write accesses will operate as $t_{RCD} = 3t_{CK}$ regardless of the setting of the RAS-CAS Delay bit (SDCTR.RCD) of the SDRAM Timing Register. The RAS-CAS Delay bit setting becomes valid when Slow Write Burst access is invalid. The setting of the Slow Burst bit does not have any effect on Read access.

9.3.9 Clock Feedback

When performing Read access at fast rates like 100 MHz, there may be insufficient set up time if an attempt to directly latch Read data with the internal clock is made. With the TX4938, it is possible to latch data using SDRAM clock SDCLKIN that is input from outside the chip. Please connect SDCLKIN to one of the SDCLK[3:0] pins and the external source.

9.3.10 ECC

9.3.10.1 ECC/Parity Mode

Table 9.3.5 shows the supported ECC/Parity functions. The ECC/Parity mode can be set separately for each channel using the ECC/Parity Mode field (SDCCRn.ECC) of the SDRAM Channel Control Register. The ECC enable bit (ECCCR.ECCE) of the ECC Control Register must be set in order to use the ECC function. No error detection, logging, or notification will be performed if this bit is not set.

Table 9.3.5 ECC/Parity Mode

ECC Field	Mode Name	Description
0x0	NOP Mode	Disables the ECC/Parity function.
0x1	EC Mode	EC (Error Check) enable Read: Performs only error checking. Correction is not performed. Write: Generates check code.
0x2	ECC Mode	ECC (Error Check and Correct) enable Read: Performs error checking and correction. Write: Generates check code.
0x3	ECC + Scrub Mode	ECC + scrub enable Read: Performs error checking and correction. Corrected data is written back to memory if an error occurs. Write: Generates check code.
0x4	Even Parity Mode	Even parity enable Read: Performs error checking. Write: Generates even parity.
0x5	Odd Parity Mode	Odd parity enable Read: Performs error checking. Write: Generates odd parity.
0x6	—	Reserved
0x7	—	Reserved

- The ECC/Parity Mode changes dynamically according to each channel setting.
- Error checking is performed when writing data smaller than 64 bits when Memory Read access is being performed while in the EC Mode, ECC Mode, or ECC + scrub mode.
- Data correction is performed if the read data cause a single-bit error when in the ECC Mode or the ECC + scrub mode. Data is read unchanged when in any other mode regardless of whether or not an error occurs.

9.3.10.2 ECC Error Notification

When either an ECC error or a parity error occurs, error data is written into one of the following fields, then error notification is performed as described below:

- Error Address Field (ERRAD) in the ECC Status Register (ECCSR)
- Error ECC/Parity Mode Field (ERRMODE)
- Error Memory Width Field (ERRMW)
- Error Syndrome Field (ERRS)

The Multi-bit Error bit (ECCSR.MBERR) of the ECC Status Register is set and an interrupt is generated if either an ECC multi-bit error or parity error is detected during any Read/Write access while the Multi-bit Error Interrupt Enable bit (ECCCR.MEI) is set.

The Single-bit Error bit (ECCSR.SBERR) of the ECC Status Register is set and an interrupt is generated if an ECC single-bit error is detected during any Read/Write access while the Single-bit Error Interrupt Enable bit (ECCCR.SEI) is set.

Multi-bit errors are assigned a higher priority than single-bit errors. If a multi-bit error is detected while the Single-bit Error bit (ECCSR.SBERR) is set, then the Single-bit Error bit (ECCSR.SBERR) is cleared, error data is written for the multi-bit error, then error notification is performed. If a single-bit error is detected while the Multi-bit Error bit (ECCSR.MBERR) is set, the Single-bit Error bit (ECCSR.SBERR) is not set and not error data is written. However, the single-bit error is corrected according to the usual procedure.

The following error notification will also be performed if either an ECC multi-bit error or parity error is detected while the Multi-bit Error Bus Error Enable Bit (ECCCR.MEB) of the ECC Control Register is set.

During read access by the TX49 core, bus error notification is sent to the TX49 core and an exception is generated. A nonmaskable interrupt is generated during Read-Modify-Write memory Read access that is performed when writing from the TX49 core data that is smaller than 64 bits. Bus error notification is sent to the appropriate bus master during Read/Write access from another bus master.

9.3.10.3 Adding Read Latency for Each ECC/Parity Mode

When using the ECC/parity function, memory access latency is added according to which ECC/parity mode is selected, whether errors will be generated or not, the error type to be generated, and whether or not to generate bus errors. Table 9.3.6 shows in cycles the memory Read access latency that will be added based on NOP mode operation under each condition.

Table 9.3.6 Read Latency Added for Each ECC/Parity Mode

ECC/Parity Mode	Bus Error Notification (ECCCR.MEB)	Error Type/Operation	Added Read Latency (in cycles)
NOP Mode	—	—	0
EC Mode	Disable	No error	0
		SBErr: Do not correct MBErr: Correct	0
	Enable	No error	1
		SBErr: Do not correct MBErr: Do not correct	2
ECC Mode	Disable	No error	1
		SBErr: Correct MBErr: Do not correct	2
	Enable	No error	1
		SBErr: Correct MBErr: Do not correct	2
ECC + scrub Mode	Disable	No error	1
		SBErr: Correct & scrub	Max. 22
		MBErr: Do not correct	3
	Enable	No error	1
		SBErr: Correct & scrub	Max. 22
		MBErr: Do not correct	2
Even Parity Mode Odd Parity Mode	Disable	No error	0
		MBErr: Do not correct	0
	Enable	No error	1
		MBErr: Do not correct	1

SBErr = Single-bit error
MBErr = Multi-bit error

9.3.10.4 ECC Memory Access

8-bit check code is used whether the data bus width is 64 bits or 32 bits. For 32-bit data bus width, check code is generated for and the error check is performed on 64-bit data that consists of two 32-bit data at the double word boundary.

CB[7:0] are used for check code reading and writing when in the 64-bit mode. CB[3:0] are used for check code reading and writing when in the 32-bit mode. An 8-bit check code is used for 64-bit data. Similar to the data however, accesses to the memory are divided into half with 4 bits being accessed at a time. The upper 4 bits of the check code are accessed simultaneous to when the upper 32 bits of the data are written or read. Similarly, the lower 4 bits of the check code are accessed simultaneous to when the lower 32 bits of the data are written or read.

All 64-bit data are always read and checked when set in the EC mode, ECC mode, or ECC + Scrub mode. Consequently, data at the double word boundary, including this data, is read and checked even when accessing data smaller than a double word (word access, byte access, etc.).

Read-Modify-Write (RMW) is performed during a Write operation of less than a double word. First, 64 bits of data that include the address where the writing is performed is read. Then, check code is generated for new 64-bit data that has replaced the written data.

Single-bit errors are corrected, but multi-bit errors are not. Therefore, if multi-bit errors are detected, no data are written back to memory.

If data is being transferred between external I/O and memory during a DMAC single address transfer, check code will not be generated even if the ECC function has been enabled.

9.3.10.5 Diagnostic Mode

Setting the Diagnostic Mode bit (ECCCR.DM) of the ECC Control Register makes it possible to use the Diagnostic Mode. When in this mode and writing to a channel for which the ECC function is enabled, the code that is set in the Diagnostic ECC field (ECCCR.DECC) is written in place of the code that was calculated from the Write data.

9.4 Registers

Table 9.4.1 SDRAM Control Register

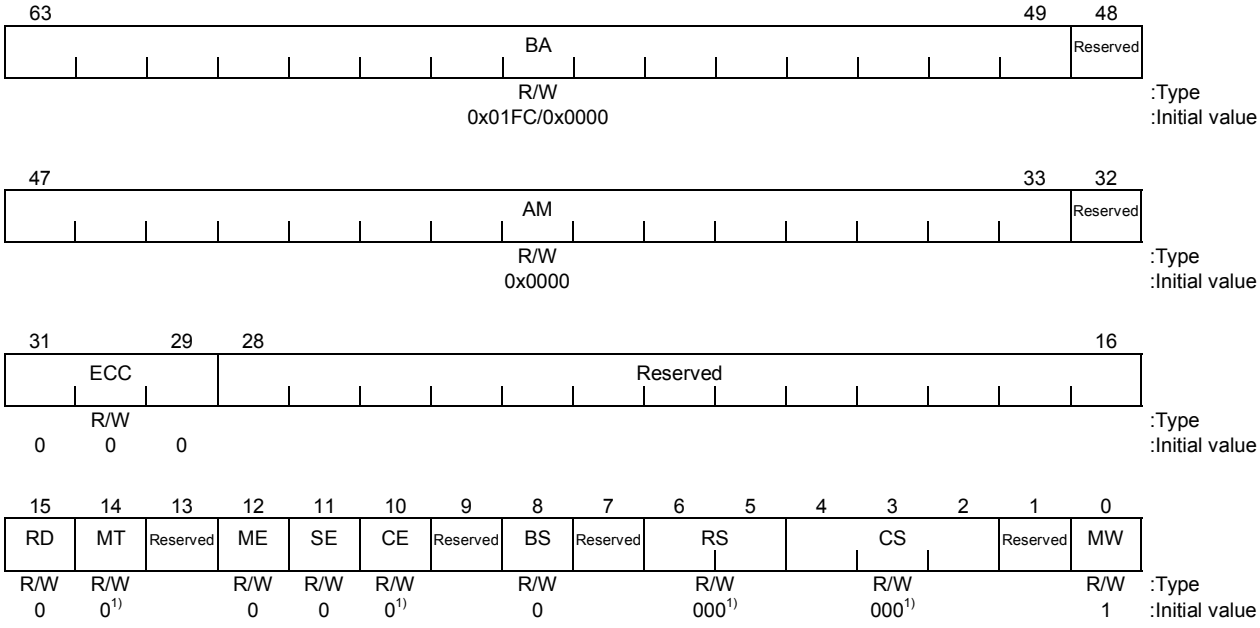
Offset Address	Bit Width	Register Symbol	Register Name
0x8000	64	SDCCR0	SDRAM Channel Control Register 0
0x8008	64	SDCCR1	SDRAM Channel Control Register 1
0x8010	64	SDCCR2	SDRAM Channel Control Register 2
0x8018	64	SDCCR3	SDRAM Channel Control Register 3
0x8040	64	SDCTR	SDRAM Timing Register
0x8058	64	SDCCMD	SDRAM Command Register

Table 9.4.2 ECC Control Register

Offset Address	Bit Width	Register Symbol	Register Name
0xA000	64	ECCCR	ECC Control Register
0xA008	64	ECCSR	ECC Status Register

9.4.1 SDRAM Channel Control Register (SDCCRn) 0x8000 (ch. 0)
0x8008 (ch. 1)
0x8010 (ch. 2)
0x8018 (ch. 3)

When the SDCCRn is programmed using a sequence of 32-bit store instructions, the base address and the address mask in the high-order 32-bit portion of the register must be written first, followed by the Channel Enable bit in the low-order 32-bit portion.



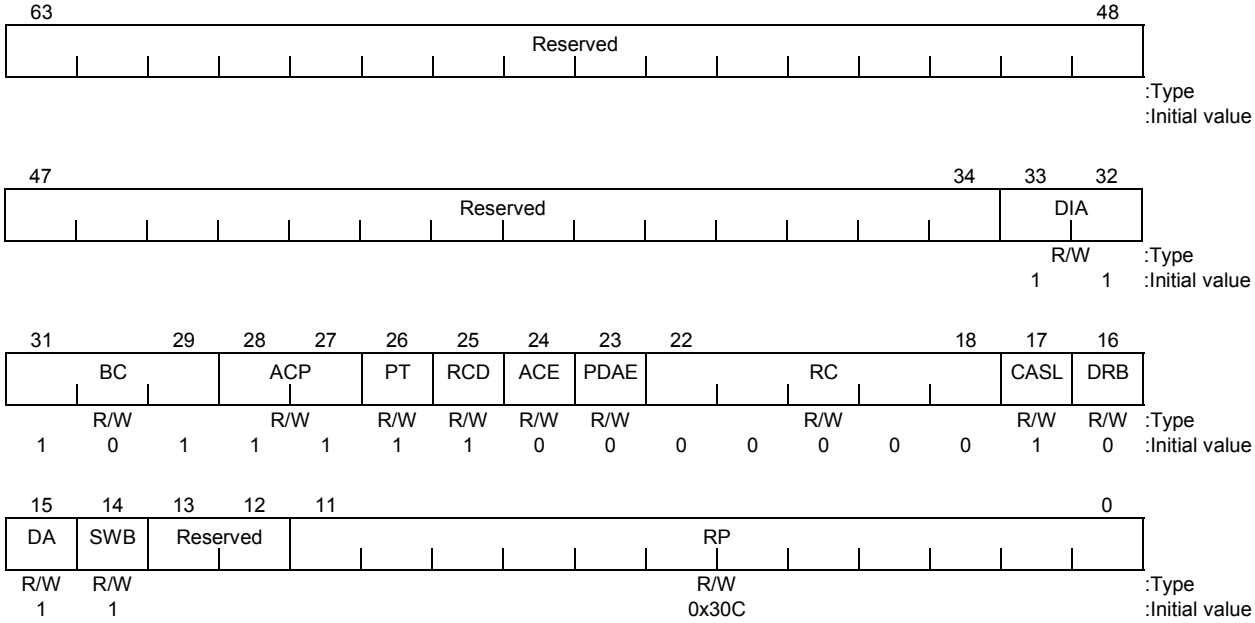
Bit	Mnemonic	Field Name	Description	Read/Write
63:49	BA[35:21]	Base Address	Base Address (Default: 0x01FC/0x0000) Specifies the base address. The upper 15 bits [35:21] of the physical address are compared to the value of this field. (Note) Only the default for Channel 0 differs. Channel 0: 0x01FC, Others: 0x0000	R/W
48	—	—	Reserved	—
47:33	AM[35:21]	Address Mask	Address Mask (Default: 0x0000) Sets the valid bits for address comparison according to the base address. 0: Bits of the corresponding BA field are compared. 1: Bits of the corresponding BA field are not compared.	R/W
32	—	—	Reserved	—
31:29	ECC	ECC/Parity Mode	ECC/Parity mode (Default: 000) Specifies the channel ECC/Parity type (refer to 9.3.10.1). 000: NOP Mode 001: EC Mode 010: ECC Mode 011: ECC + scrub Mode 100: Even Parity Mode 101: Odd Parity Mode 110: Reserved 110: Reserved	R/W
28:16	—	—	Reserved	—

Figure 9.4.1 SDRAM Channel Control Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
15	RD	Registered DIMM	Registered DIMM (Default: 0) Specifies whether the SDRAM connected to the channel is Registered memory. 0: Disable Registered memory 1: Enable Registered memory	R/W
14:13	—	—	Reserved	—
12	ME	Master Enable	Master Enable (Default: 0) Specifies during ECC initialization whether a channel will be made the Master channel. 0: Disable 1: Enable	R/W
11	SE	Slave Enable	Slave Enable (Default: 0) Specifies during ECC initialization whether a channel will be made the Slave channel. 0: Disable 1: Enable	R/W
10	CE	Channel Enable	Enable (Default: 0) Specifies whether to enable a channel. 0: Disable 1: Enable	R/W
9	—	—	Reserved	—
8	BS	Bank Count	Number of Banks (Default: 0) Specifies the bank count. 0: 2 banks 1: 4 banks	R/W
7	—	—	Reserved	—
6:5	RS	Row Size	Row Size (Default: 00) Specifies the row size. 00: 2048 Rows (11 bits) 01: 4096 Rows (12 bits) 10: 8192 Rows (13 bits) 11: Reserved	R/W
4:2	CS	Column Size	Column Size (Default: 000) Specifies the column size. 000: 256 words (8 bits) 001: 512 words (9 bits) 010: 1024 words (10 bits) 011: 2048 words (11 bits) 100: 4096 words (12 bits) 101-111: Reserved	R/W
1	—	—	Reserved	—
0	MW	Memory Width	Memory Width (Default: 0) Specifies the bus width. 0: 64 bits 1: 32 bits	R/W

Figure 9.4.1 SDRAM Channel Control Register (2/2)

9.4.2 SDRAM Timing Register (SDCTR) 0x8040



Bit	Mnemonic	Field Name	Description	Read/Write
63:34	—	—	Reserved	—
33:32	DIA	Write Active Period	Data In to Active(t_{DAL}) (Default: 11) Specifies the period from the last Write data to the Active command. 00: Reserved[1] 01: 4 t_{CK} 10: 5 t_{CK} 11: 6 t_{CK}	R/W
31:29	BC	Bank Cycle Time	Bank Cycle Time (t_{RC}) (Default: 101) Specifies the bank cycle time.[2] 000: 5 t_{CK} 100: 9 t_{CK} 001: 6 t_{CK} 101: 10 t_{CK} 010: 7 t_{CK} 110: Reserved 011: 8 t_{CK} 111: Reserved	R/W
28:27	ACP	Active Command Time	Active Command Period (t_{RAS}) (Default: 11) Specifies the active command time. 00: 3 t_{CK} 01: 4 t_{CK} 10: 5 t_{CK} 11: 6 t_{CK}	R/W
26	PT	Precharge Time	Precharge Time (t_{RP}) (Default: 1) Specifies the precharge time. 0: 2 t_{CK} 1: 3 t_{CK}	R/W
25	RCD	RAS-CAS Delay	RAS to CAS Delay (t_{RCD}) (Default: 1) Specifies the RAS - CAS delay. 0: 2 t_{CK} 1: 3 t_{CK}	R/W

Figure 9.4.2 SDRAM Timing Register (1/2)

[1] t_{CK} = Clock cycle

[2] t_{RC} is used during (i) refresh cycle time, (ii) single Read, (iii) two transfer burst Reads. The bank cycle time is t_{RAS} + t_{RP} + 1t_{CK} if t_{RAS} + t_{RP} < t_{RC} in the case of (ii) (iii).

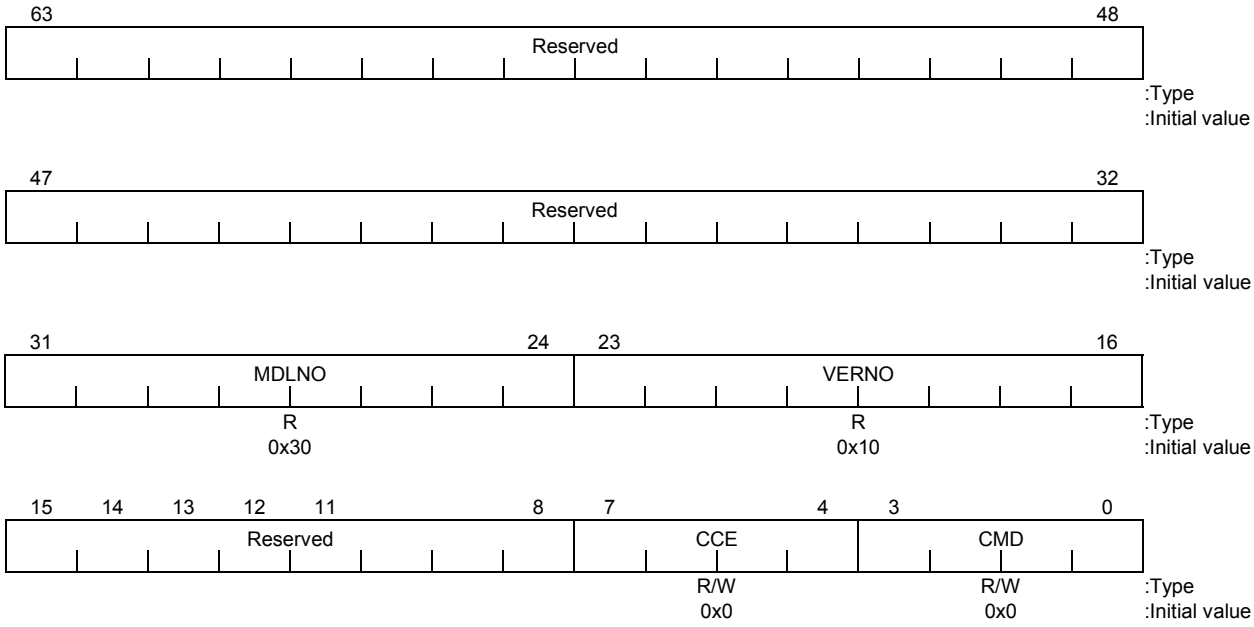
Bit	Mnemonic	Field Name	Description	Read/Write
24	ACE	Advanced CKE	Advanced CKE enable (Default: 0) Enabling this function makes the timing at which CKE changes one cycle earlier. 0: Disable 1: Enable	R/W
23	PDAE	Power Down Auto Entry	Power Down Auto Entry Enable (Default: 0) Enabling this function makes CKE become “L” while the SDRAMC is in the Idle state. When refresh, memory access, or command execution is performed, CKE automatically becomes “H”, the requested operation is performed, then CKE returns to “L” when the operation is complete. 0: Disable 1: Enable	R/W
22:18	RC	Refresh Counter	Refresh Counter (Default: 000000) This counter is decremented at each refresh. If the refresh circuit is activated and a value other than “0” is loaded, this field becomes a down counter that stops at “0”. A value other than “0” must be reloaded to start the countdown again. This is used during memory initialization.	R/W
17	CASL	CAS Latency	CAS Latency (t_{CASL}) (Default: 1) Specifies the CAS latency. 0: 2 t_{CK} 1: 3 t_{CK}	R/W
16	DRB	Data Read Bypass	Data Read Bypass (Default: 0) Selects the Data Read path used. 0: Data Read latches to the register using the feedback clock. 1: Data Read bypasses the feedback clock latch.	R/W
15	DA	Active Command Delay	Delay Activate (t_{DA}) (Default: 1) Specifies the delay from the row address to the bank active command. Setting this bit to “1” sets up the row address two cycles before the active command is executed. 0: 0 t_{CK} 1: 1 t_{CK}	R/W
14	SWB	Slow Write Burst	Slow Write Burst (t_{SWB}) (Default: 1) Specifies whether to perform Slow Write Burst. 0: Burst Write occurs at each 1 t_{CK} 1: Burst Write occurs at each 2 t_{CK}	R/W
13:12	—	—	Reserved	—
11:0	RP	Refresh Period	Refresh Period (Default: 0x30c) Specifies the clock cycle count that generates the refresh cycle. Refresh is only enabled when at least one SDRAM channel is enabled. Please program the Timing Register before an arbitrary channel is enabled. Default is 0x30C. A refresh cycle occurs for each 7.8 μs@100 MHz in this situation.	R/W

Figure 9.4.2 SDRAM Timing Register (2/2)

[1] t_{CK} = Clock cycle

[2] t_{RC} is used during (i) refresh cycle time, (ii) single Read, (iii) two transfer burst Reads. The bank cycle time is t_{RAS} + t_{RP} + 1t_{CK} if t_{RAS} + t_{RP} < t_{RC} in the case of (ii) (iii).

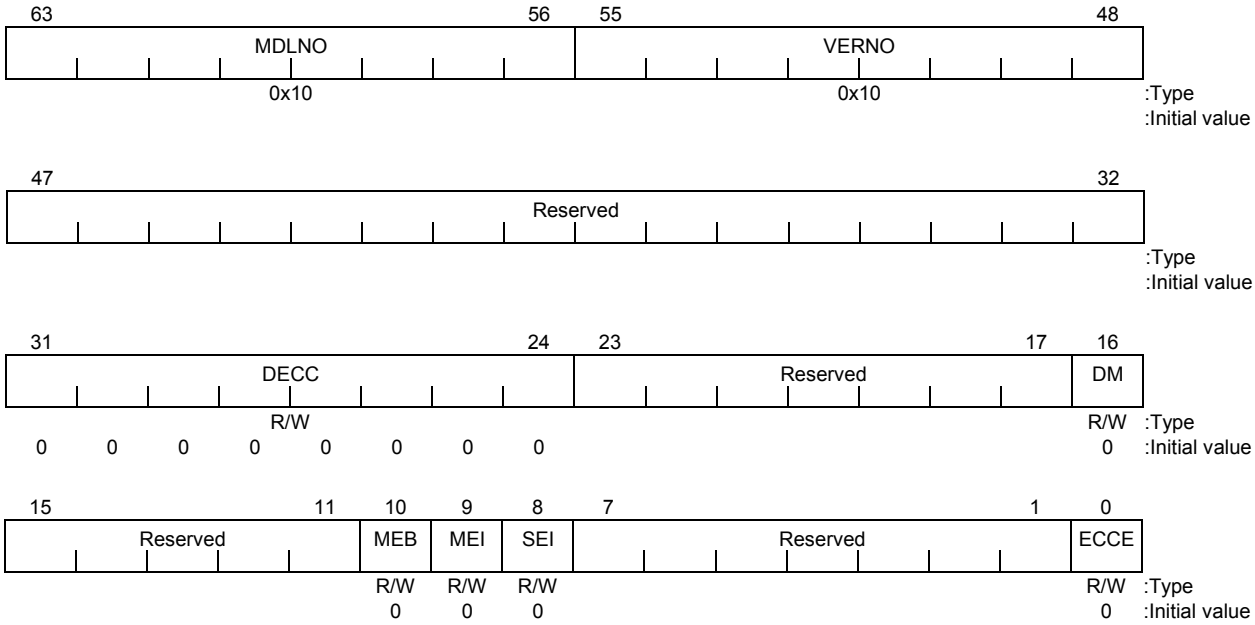
9.4.3 SDRAM Command Register (SDCCMD) 0x8058



Bit	Mnemonic	Field Name	Description	Read/Write
63:32	—	—	Reserved	—
31:24	MDLNO	Model Number	Model Number (Default: 0x30) Indicates the model number. The default value is 0x30 for the TX4938. This field is Read Only.	R
23:16	VERNO	Version Number	Version Number (Default: 0x10) Indicates the version number. The default value is 0x10 for the TX4938. This field is Read Only.	R
15:8	—	—	Reserved	—
7:4	CCE	Command Channel Enable	Command Channel Enable Setting one of these bits to “1” enables the command of the corresponding channel. This command is simultaneously executed on all channels that are enabled. bit 7: Channel 3 bit 6: Channel 2 bit 5: Channel 1 bit 4: Channel 0	R/W
3:0	CMD	Command	Command Specifies a command that is performed on memory. 0x0: NOP command 0x1: Set Mode Register command Set SDRAM Mode Register from SDCTR value 0x2: Reserved 0x3: Precharge All command Precharge All SDRAM Banks 0x4: Self-Refresh Mode command Sets SDRAM to the Self-Refresh Mode 0x5: Power Down Mode Command Set SDRAM to the Power Down Mode 0x6: Normal Mode Command Cancel Self-Refresh/Power Down Mode 0x7-0xf: Reserved	R/W

Figure 9.4.3 SDRAM Command Register

9.4.4 ECC Control Register (ECCCR) 0xA000



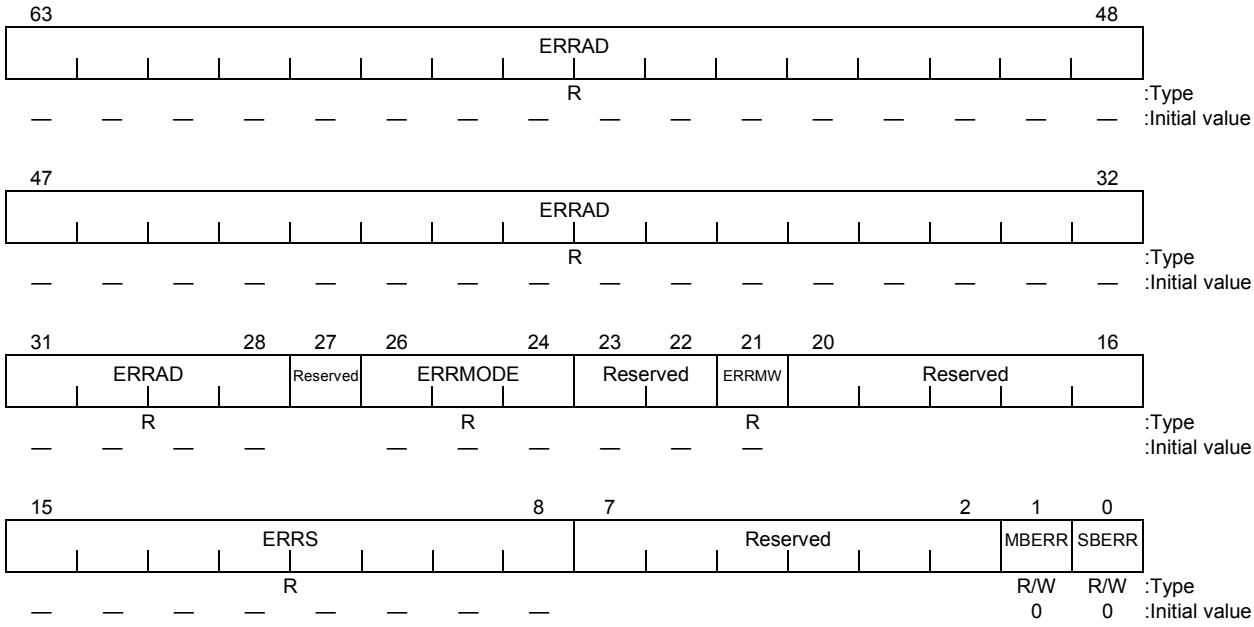
Bit	Mnemonic	Field Name	Description	Read/Write
63:56	MDLNO	Model Number	Model Number (Default: 0x10) Indicates the model number. The default value for the TX4938 is 0x10. This field is Read Only.	R
55:48	VERNO	Version Number	Address Mask (Default: 0x10) Indicates the version number. The default value for the TX4938 is 0x10. This field is Read Only.	R
47:32	—	—	Reserved	—
31:24	DECC	Diagnostic ECC	Diagnostic ECC (Default: 0x00) The value set by this field is output from CB[7:0] as the check code when the DM bit is set to “Enable.”	R/W
23:17	—	—	Reserved	—
16	DM	Diagnostic Mode	ECC Diagnostic Mode (Default: 0) Specifies whether to use the Diagnostic Mode. 0: Disable 1: Enable	R/W
15:11	—	—	Reserved	—
10	MEB	Multi-Bit Error Bus Error Enable	Multi-Bit Error Bus Error Enable (Default: 0) Specifies whether to generate a bus error when a multi-bit error occurs. When this function is enabled, an NMI is generated for RMW* errors occurring during a Write operation to the TX49/H3 core. Bus errors are generated for all other operations. 0: Disable 1: Enable	R/W
9	MEI	Multi-Bit Error Interrupt Enable	Multi-Bit Error Interrupt Enable (Default: 0) Specifies whether to generate an interrupt during a multi-bit error. 0: Disable 1: Enable	R/W

Figure 9.4.4 ECC Control Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
8	SEI	Single-Bit Error Interrupt Enable	Single-Bit Error Interrupt Enable (Default: 0) Specifies whether to generate an interrupt during a single-bit error. 0: Disable 1: Enable	R/W
7:1	—	—	Reserved	—
0	ECCE	ECC Enable	ECC Enable (Default: 0) Specifies whether to enable the ECC/Parity function. When disabled, the ECC function will not operate even if the ECC Parity Mode field (SDCCRn.ECC) selects the ECC/Parity Mode. 0: Disable 1: Enable	R/W

Figure 9.4.4 ECC Control Register (2/2)

9.4.5 ECC Status Register (ECCSR) 0xA008



Bit	Mnemonic	Field Name	Description	Read/Write
63:28	ERRAD	Error Address	Error Address (Default: Unknown) A 36-bit physical address is set when an error occurs. This address is retained until either SBERR or MBERR is cleared. This field is Read Only.	R
27	—	—	Reserved	—
26:24	ERRMODE	Error ECC/Parity Mode	Error ECC Mode (Default: Unknown) The ECC/Parity Mode is set when an error occurs. This address is retained until either SBERR or MBERR is cleared. This field is Read Only.	R
23:22	—	—	Reserved	—
21	ERRMW	Error Memory Width	Error Memory Width (Default: Unknown) The memory data width is set when an error occurs. This address is retained until either SBERR or MBERR is cleared. This field is Read Only. 0: 64 bits 1: 32 bits	R
20:16	—	—	Reserved	—
15:8	ERRS	Error Syndrome	Error Syndrome (Default: Unknown) The error syndrome for when errors occur is set. The syndrome is retained until either SBERR or MBERR is cleared. This field is Read Only.	R
7:2	—	—	Reserved	—
1	MBERR	Multi-Bit Error	Multi-Bit Error (Default: 0) This bit is set to “1” when a multi-bit error occurs, or when a parity error occurs while in the Parity Mode. Once a multi-bit error occurs, until this bit is cleared, no status in the Status Register is updated even if new multi-/single-bit errors occur. 0: No error 1: Generate error	R/W
0	SBERR	Single-Bit Error	Single-Bit Error (Default: 0) This bit is set to “1” when a single-bit error occurs. Once a single-bit error occurs, until this bit is cleared, no status in the Status Register is updated even if new single-bit error occurs. If a multi-bit error occurs, status is updated regardless of whether a single-bit error has occurred or not. 0: No error 1: Generate error	R/W

Figure 9.4.5 ECC Status Register

9.5 Timing Diagrams

Please note the following when referring to the timing diagrams in this section: the shaded area in each diagram expresses values that have yet to be determined.

9.5.1 Single Read (64-bit Bus)

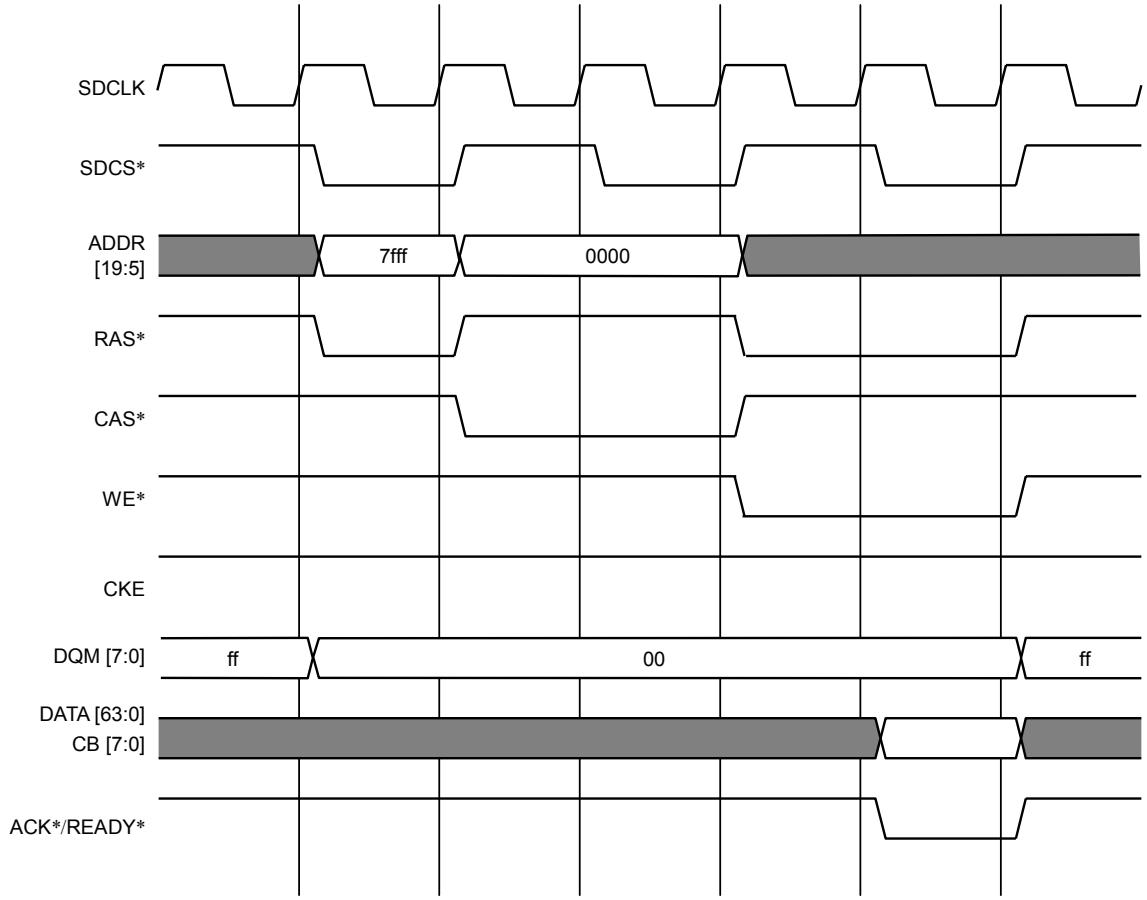


Figure 9.5.1 Single Read (t_{RCD} = 2, t_{CASL} = 2, t_{DA} = 0, 64-bit Bus)

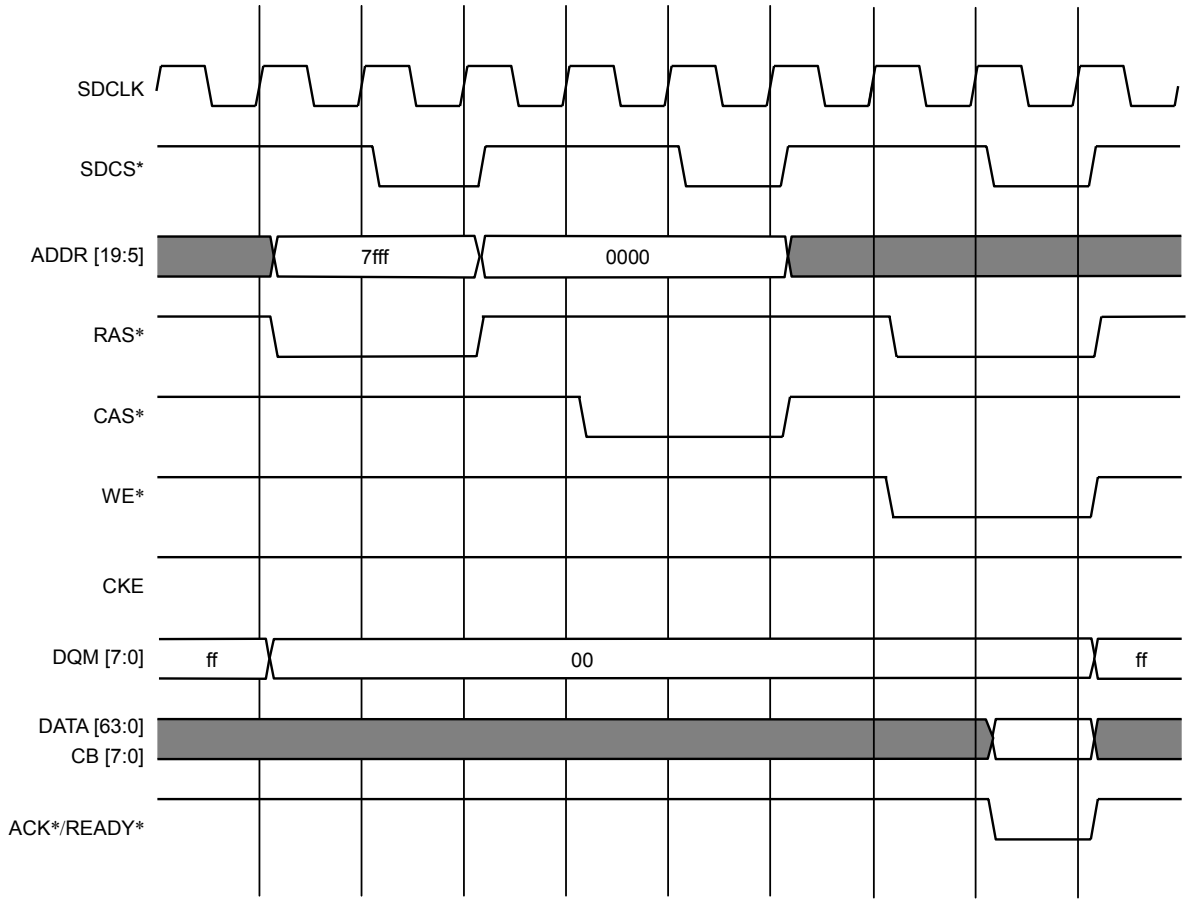


Figure 9.5.2 Single Read (t_{RCD} = 3, t_{CASL} = 3, t_{DA} = 1, 64-bit Bus)

9.5.2 Single Write (64-bit Bus)

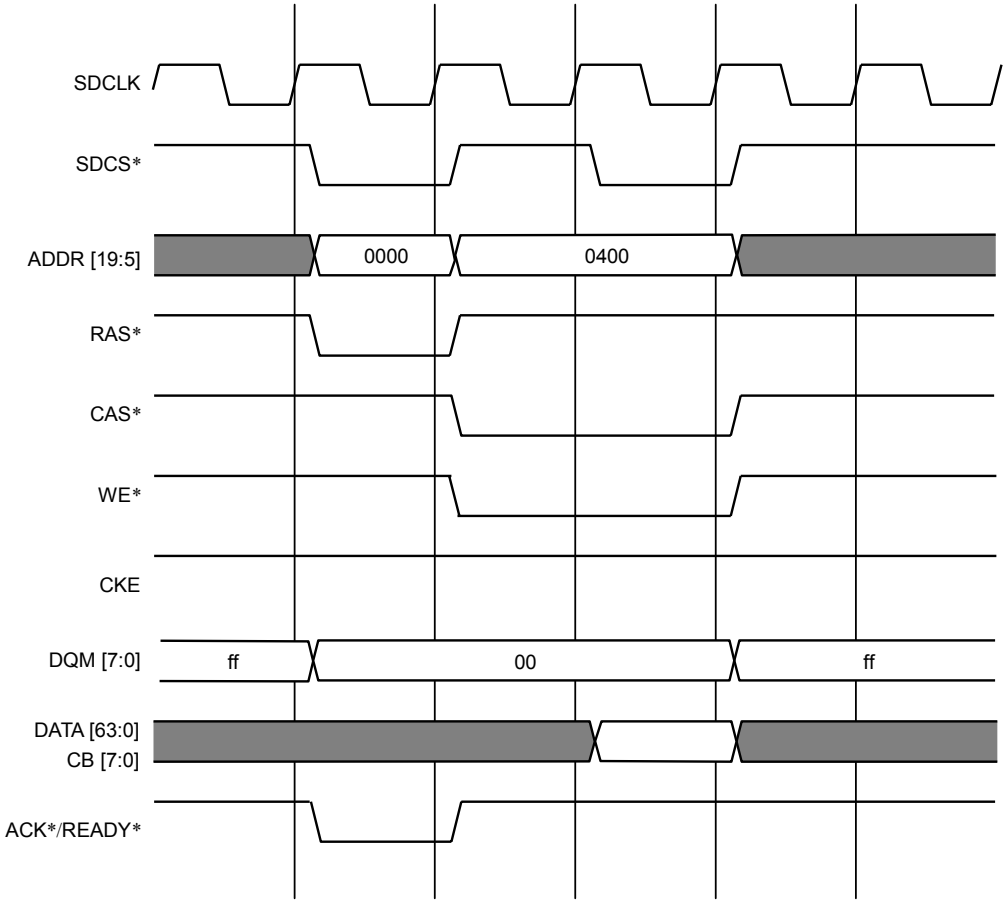


Figure 9.5.3 Double-Word Single Write (t_{RCD} = 2, t_{DA} = 0, 64-bit Bus)

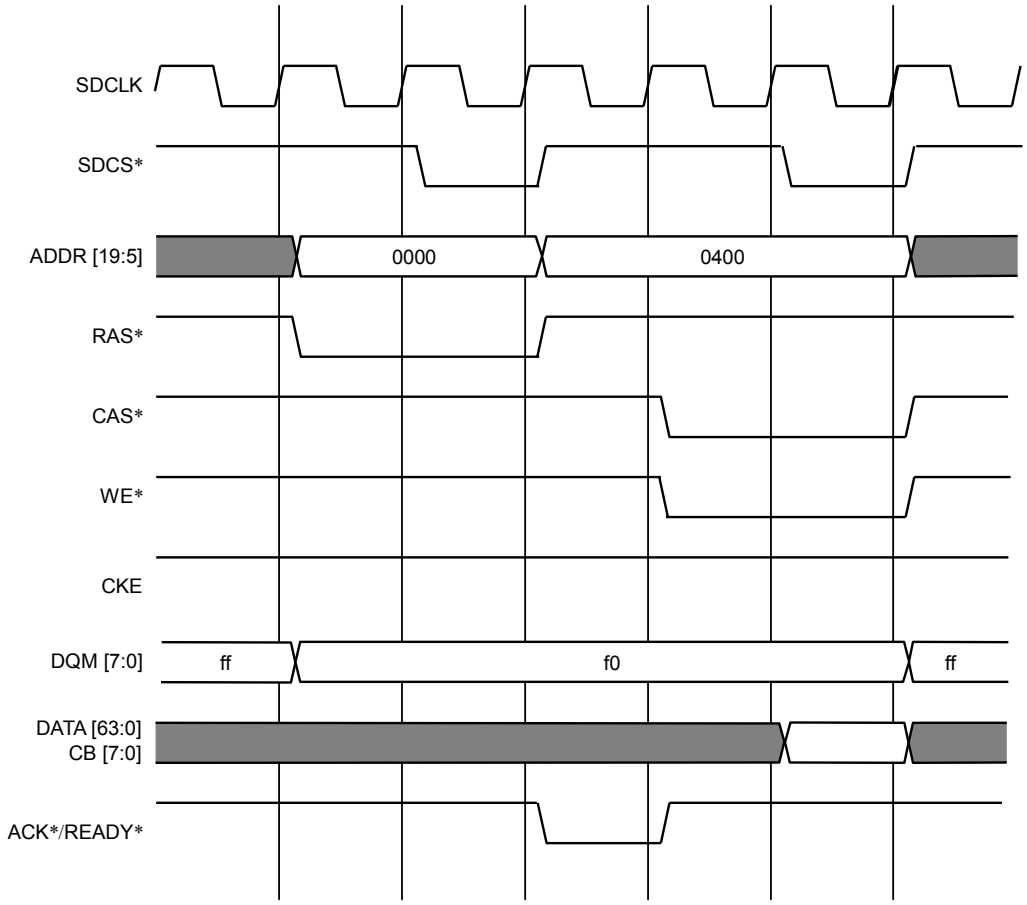


Figure 9.5.4 One-Word Single Write (t_{RCD} = 3, t_{DA} = 1, 64-bit Bus)

9.5.3 Burst Read (64-bit Bus)

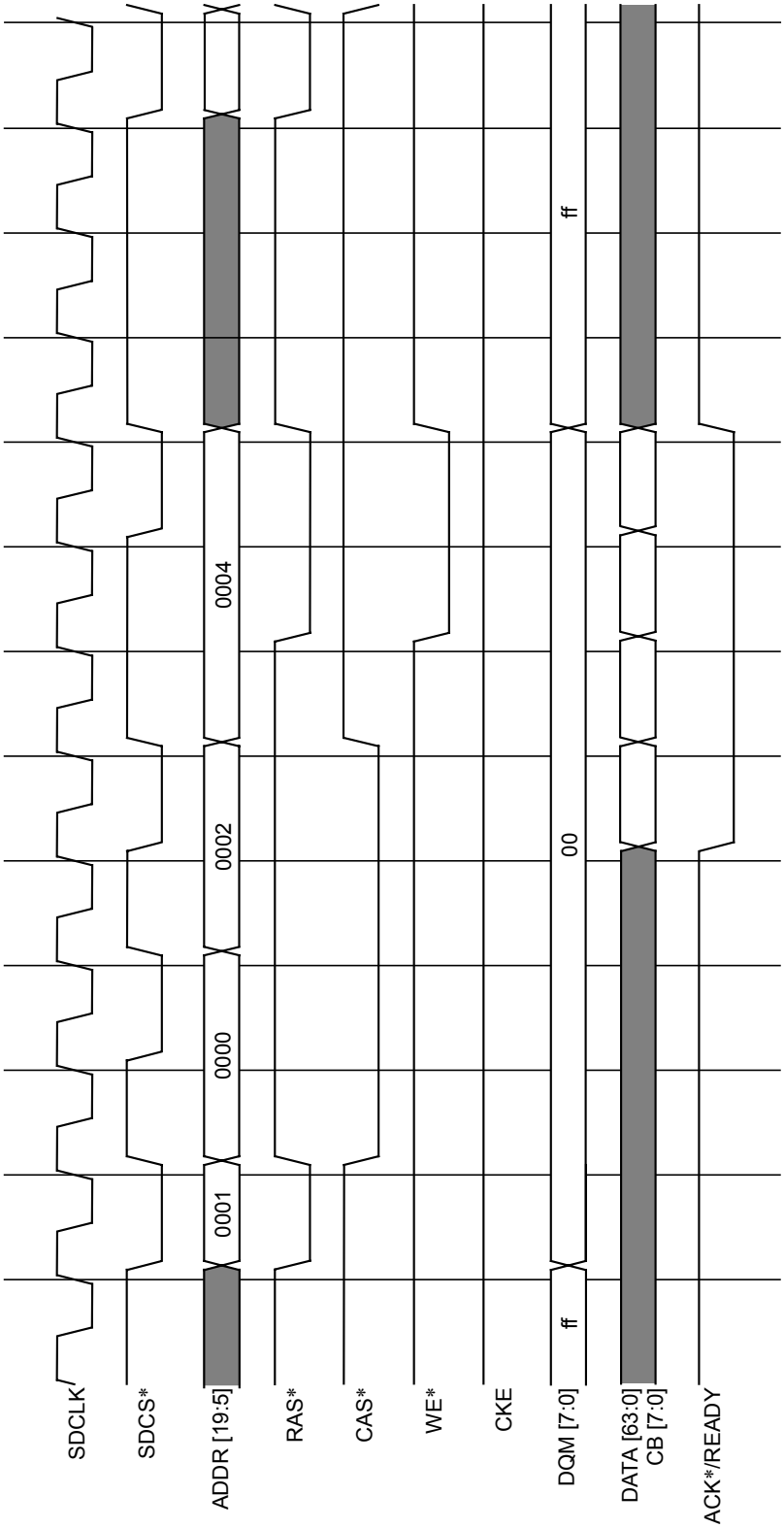


Figure 9.5.5 Eight-Word Burst Read (t_{RCD} = 2, t_{CASL} = 2, t_{DA} = 0, 64-bit Bus)

9.5.4 Burst Write (64-bit Bus)

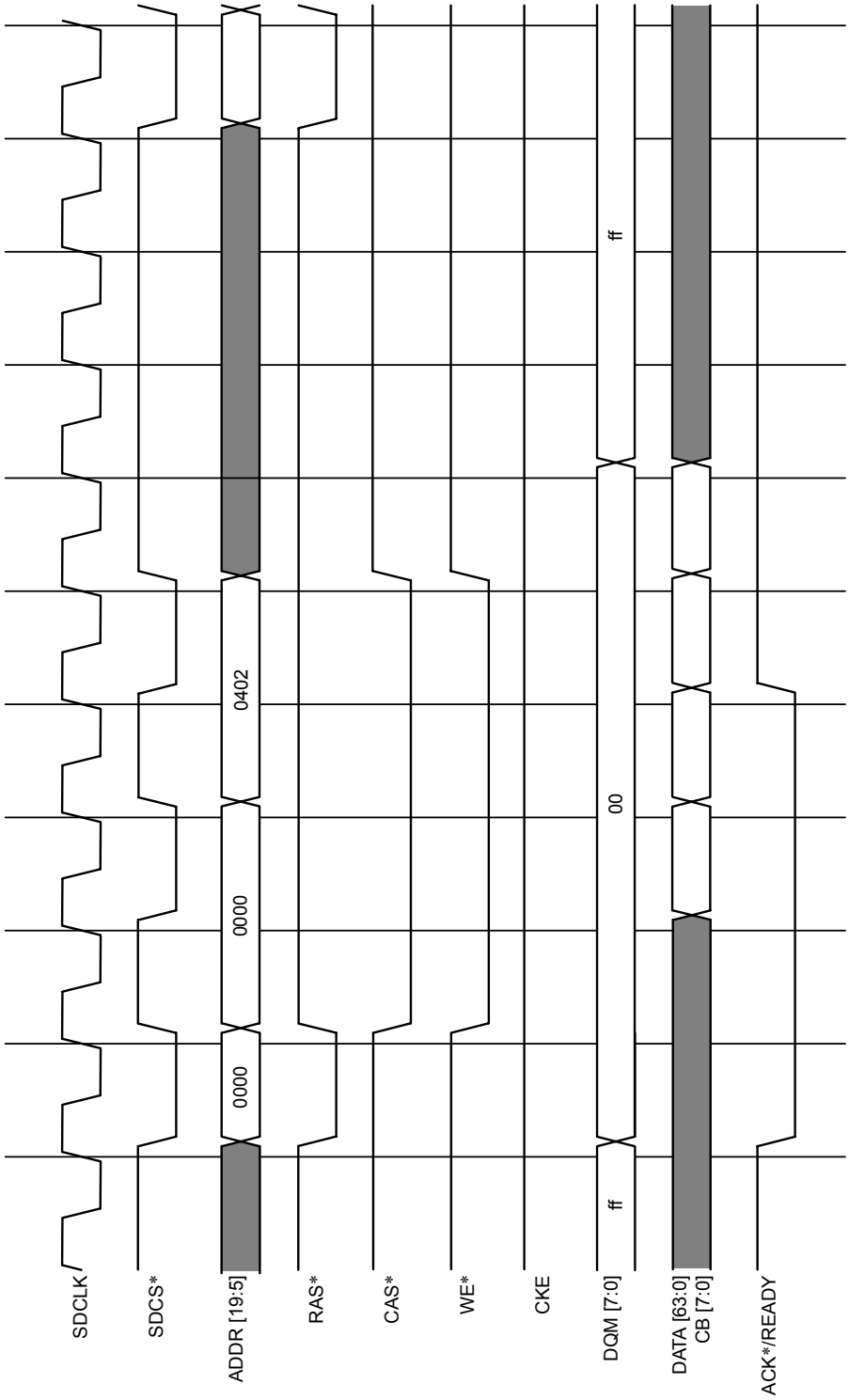


Figure 9.5.6 Eight-Word Burst Write (t_{RCD} = 2, t_{DA} = 0, 64-bit Bus)

9.5.5 Burst Write (64-bit Bus, Slow Write Burst)

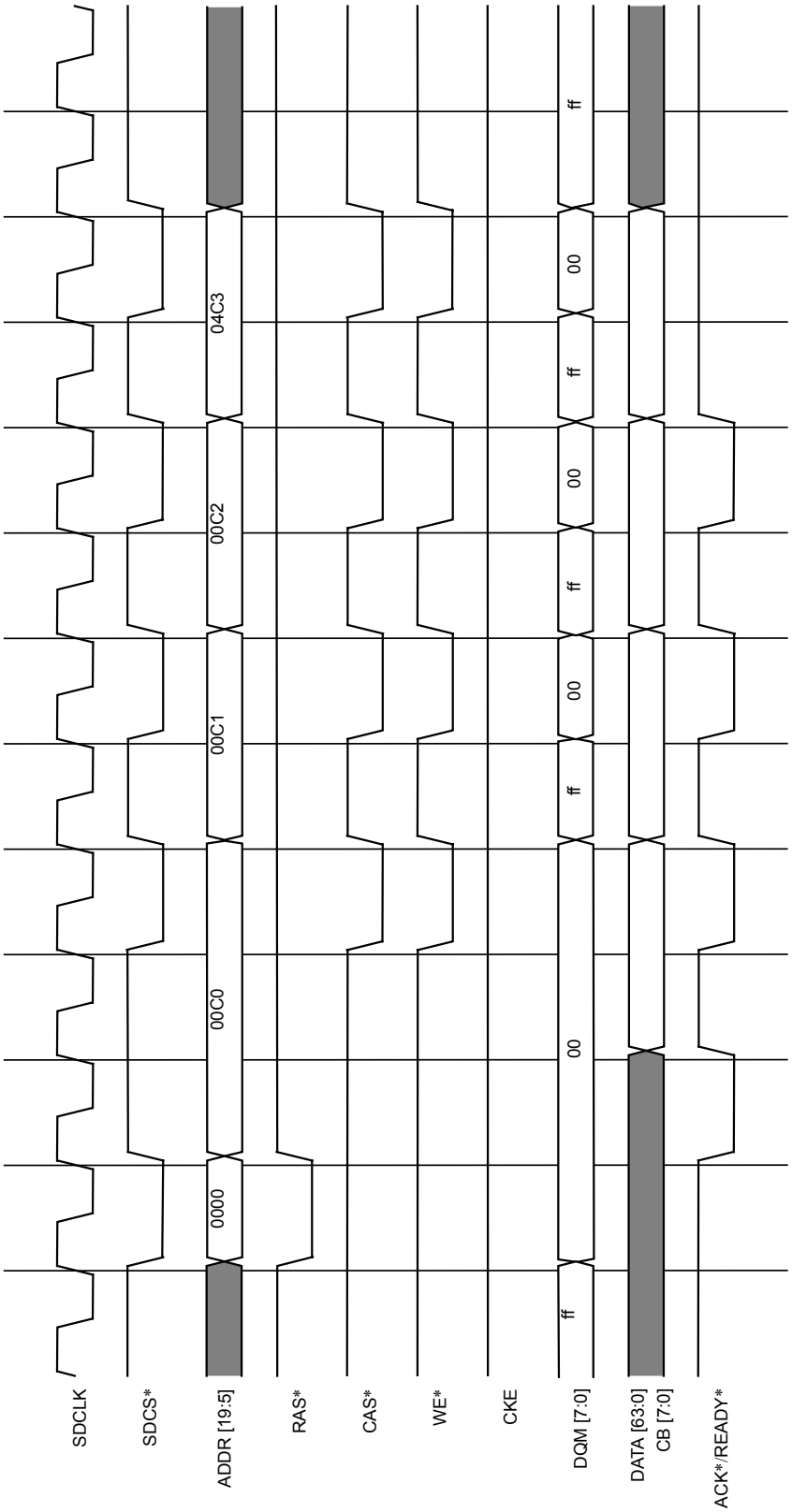


Figure 9.5.7 Eight-Word Burst Write (t_{RCD} = 2, t_{DA} = 0, 64-bit Bus, Slow Write Burst)

9.5.6 Single Read (32-bit Bus)

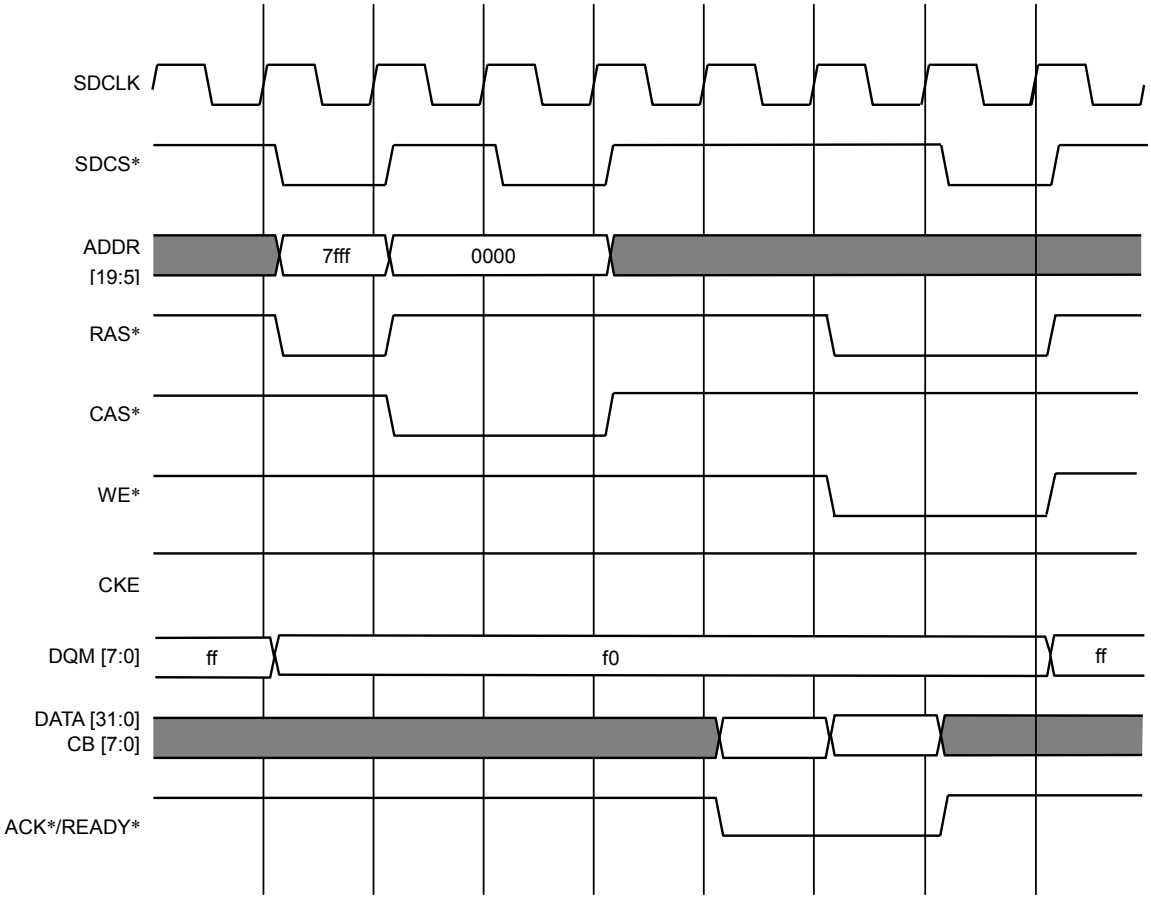


Figure 9.5.8 Double-Word Single Read (t_{RCD} = 2, t_{CASL} = 2, t_{DA} = 0, 32-bit Bus)

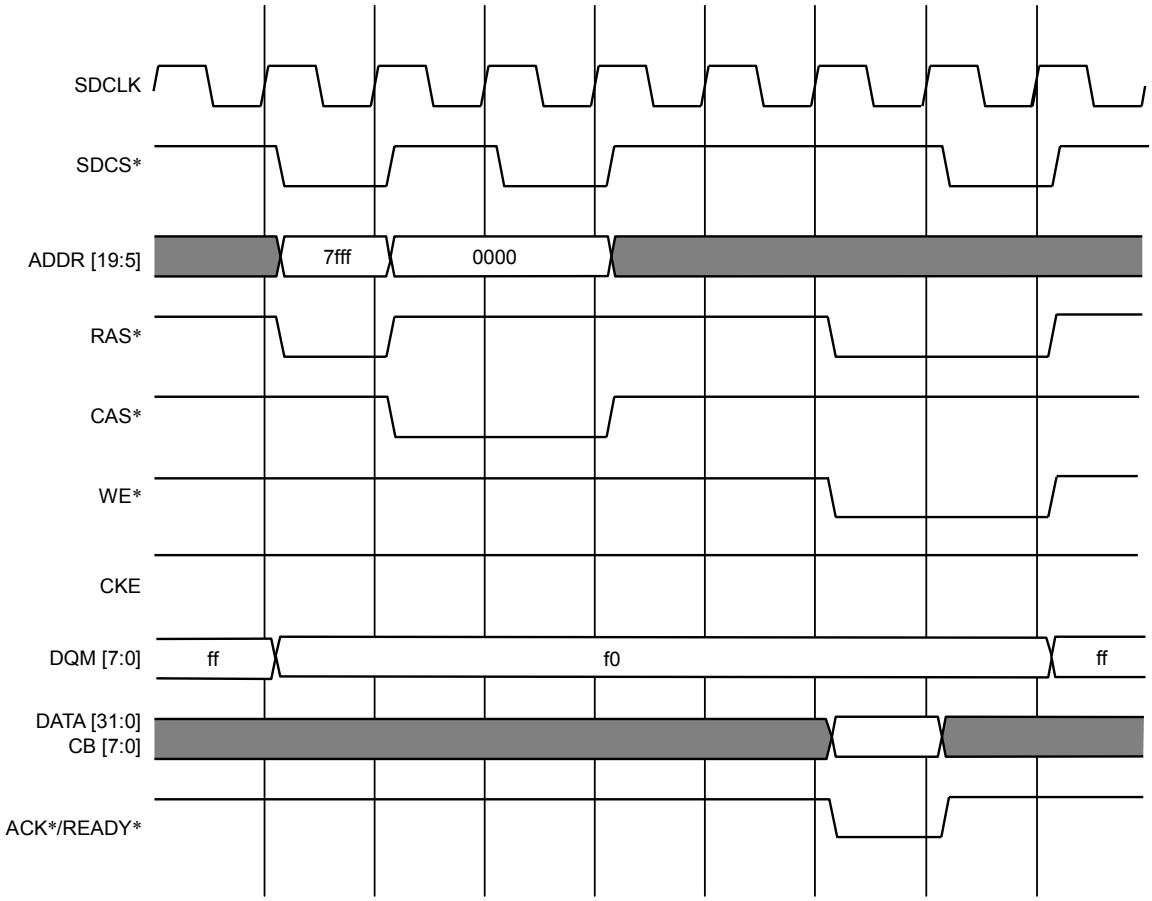


Figure 9.5.9 One-Word Single Read (t_{RCD} = 2, t_{CASL} = 3, t_{DA} = 0, 32-bit Bus)

9.5.7 Single Write (32-bit Bus)

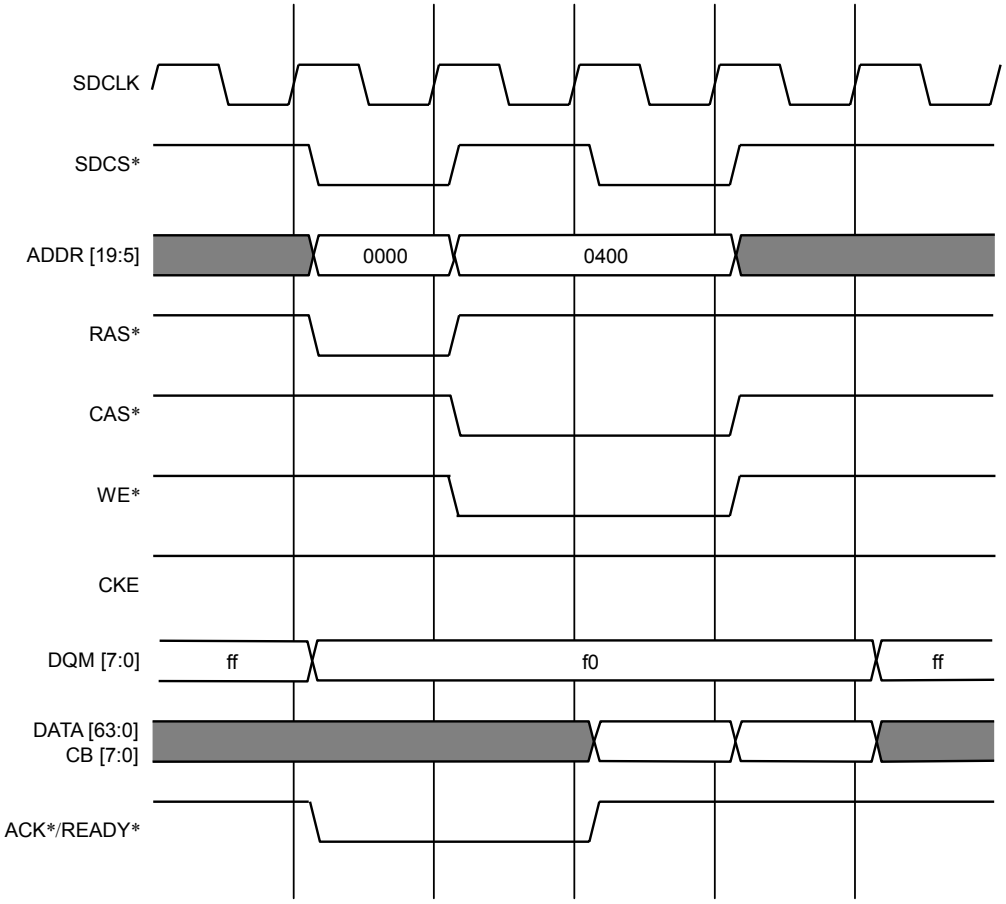


Figure 9.5.10 Double-Word Single Write (t_{RCD} = 2, t_{DA} = 0, 32-bit Bus)

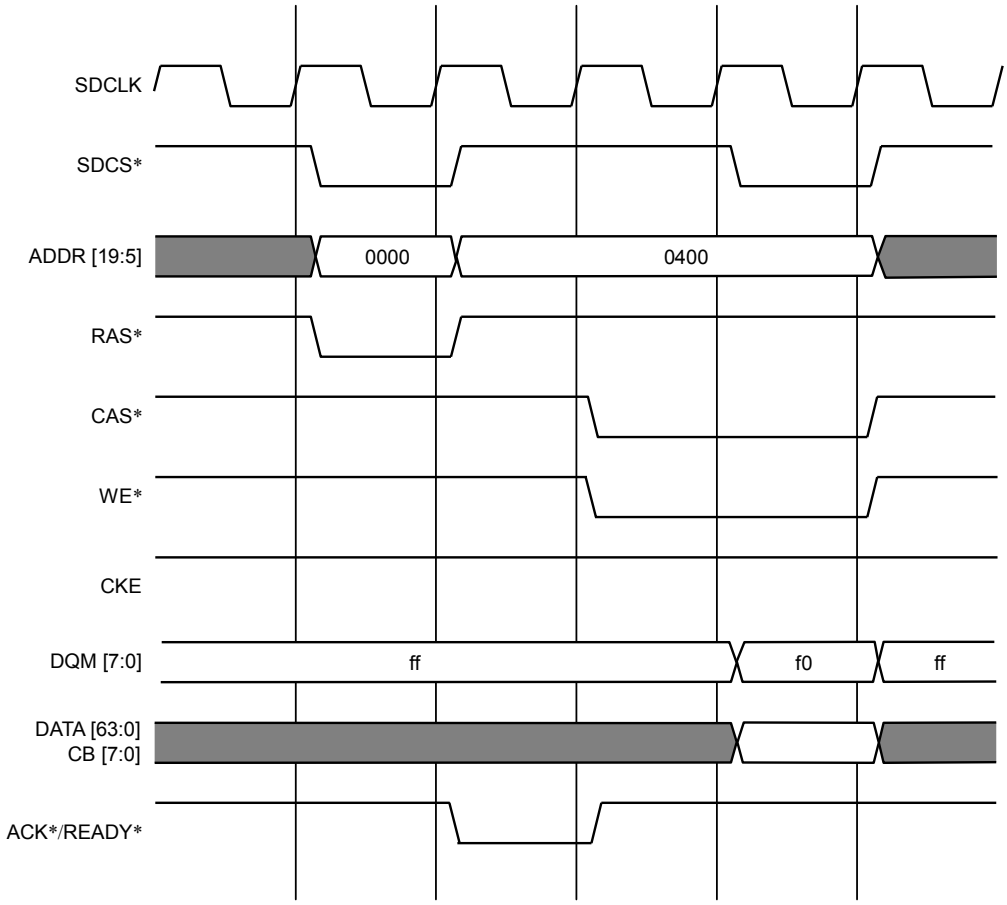


Figure 9.5.11 One-Word Single Write (t_{RCD} = 3, t_{DA} = 0, 32-bit Bus)

9.5.8 Low Power Consumption and Power Down Mode

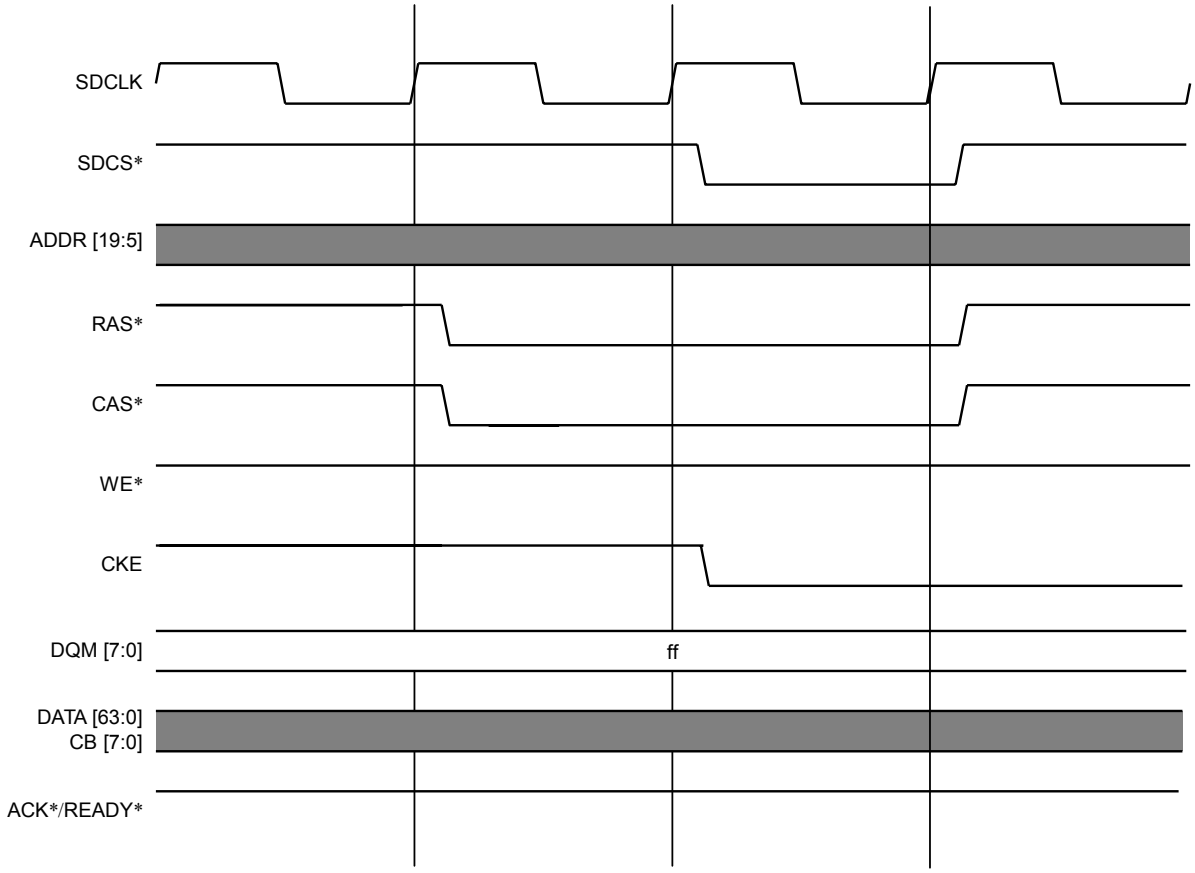


Figure 9.5.12 Transition to Low Power Consumption Mode (SDCTR.ACE = 0)

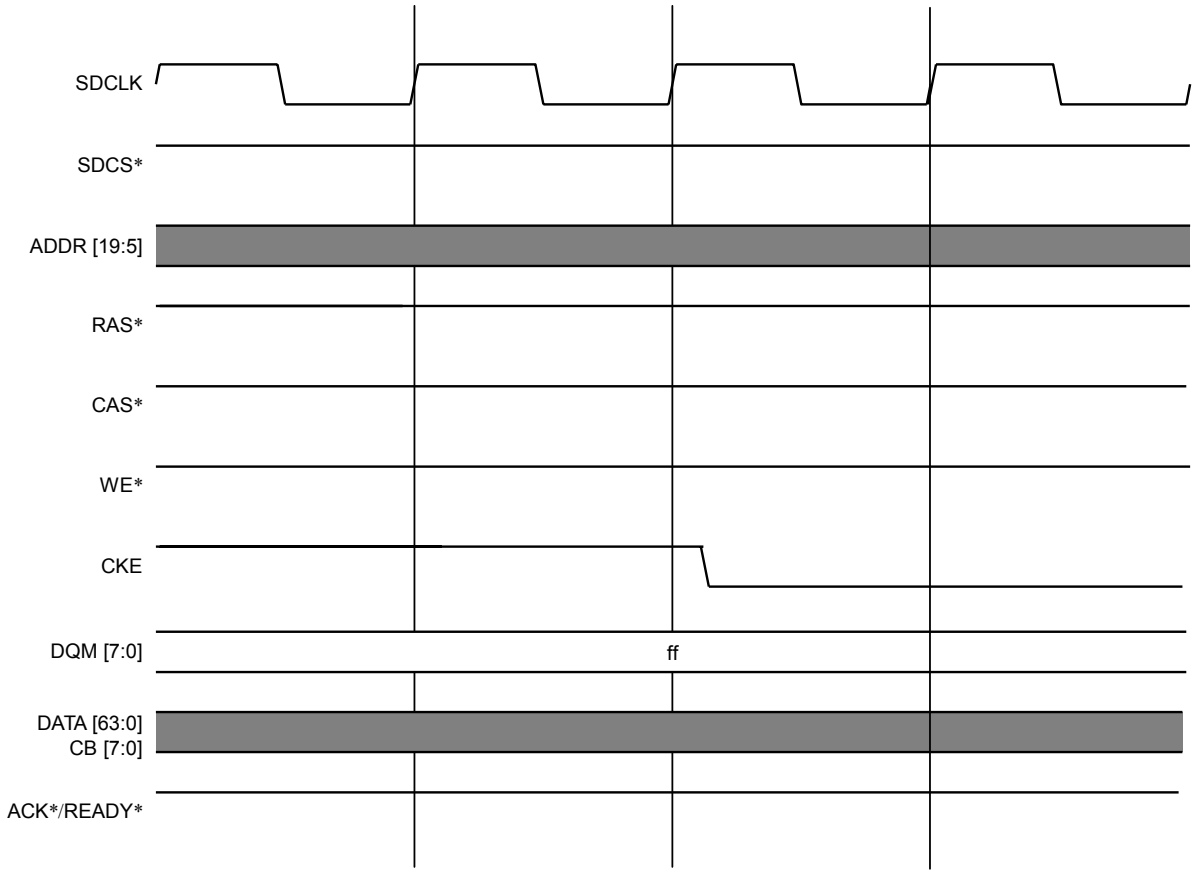


Figure 9.5.13 Transition to Power Down Mode

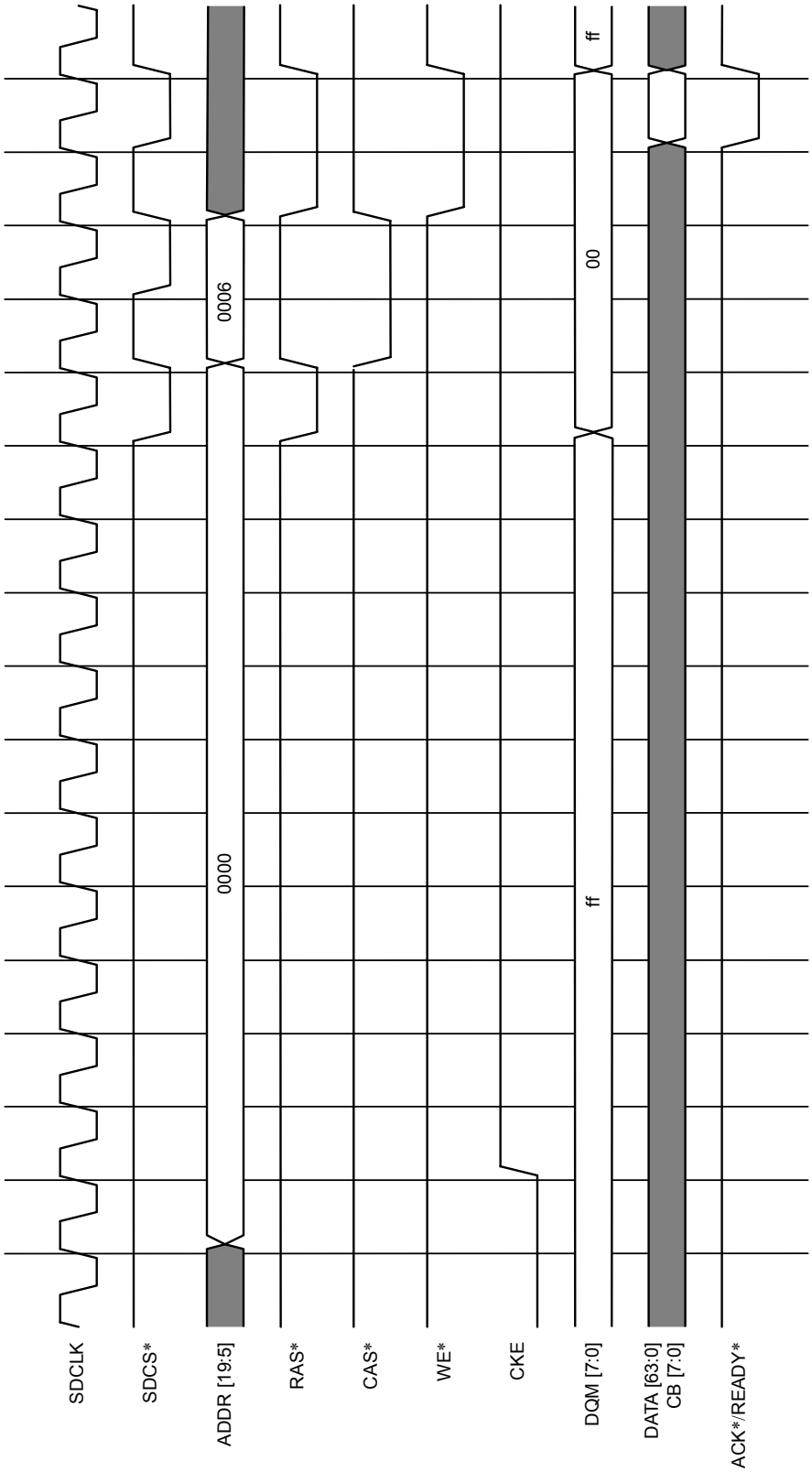


Figure 9.5.14 Return From Low Power Consumption/Power Down Mode
(SDCTR.PDAE = 0, SDCTR.ACE = 0)

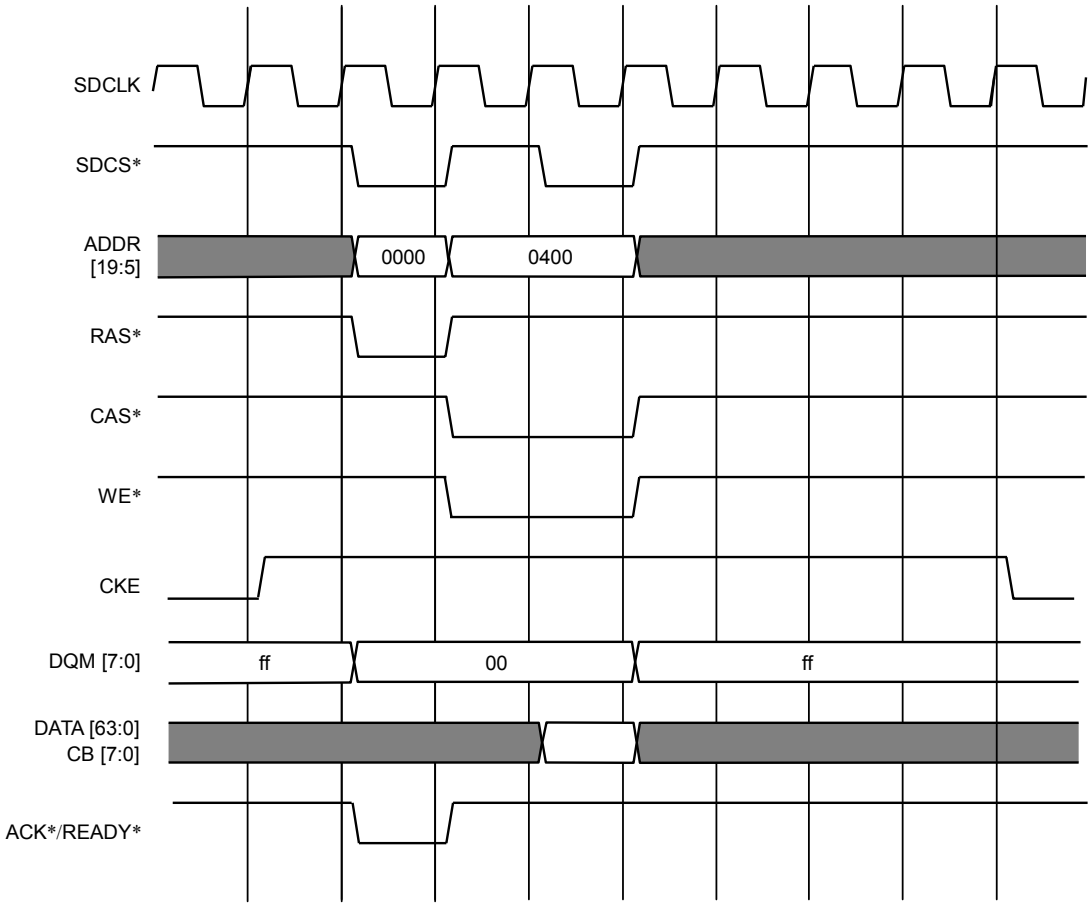


Figure 9.5.15 Power Down Auto Entry (SDCTR.PDAE = 1, SDCTR.ACE = 0)

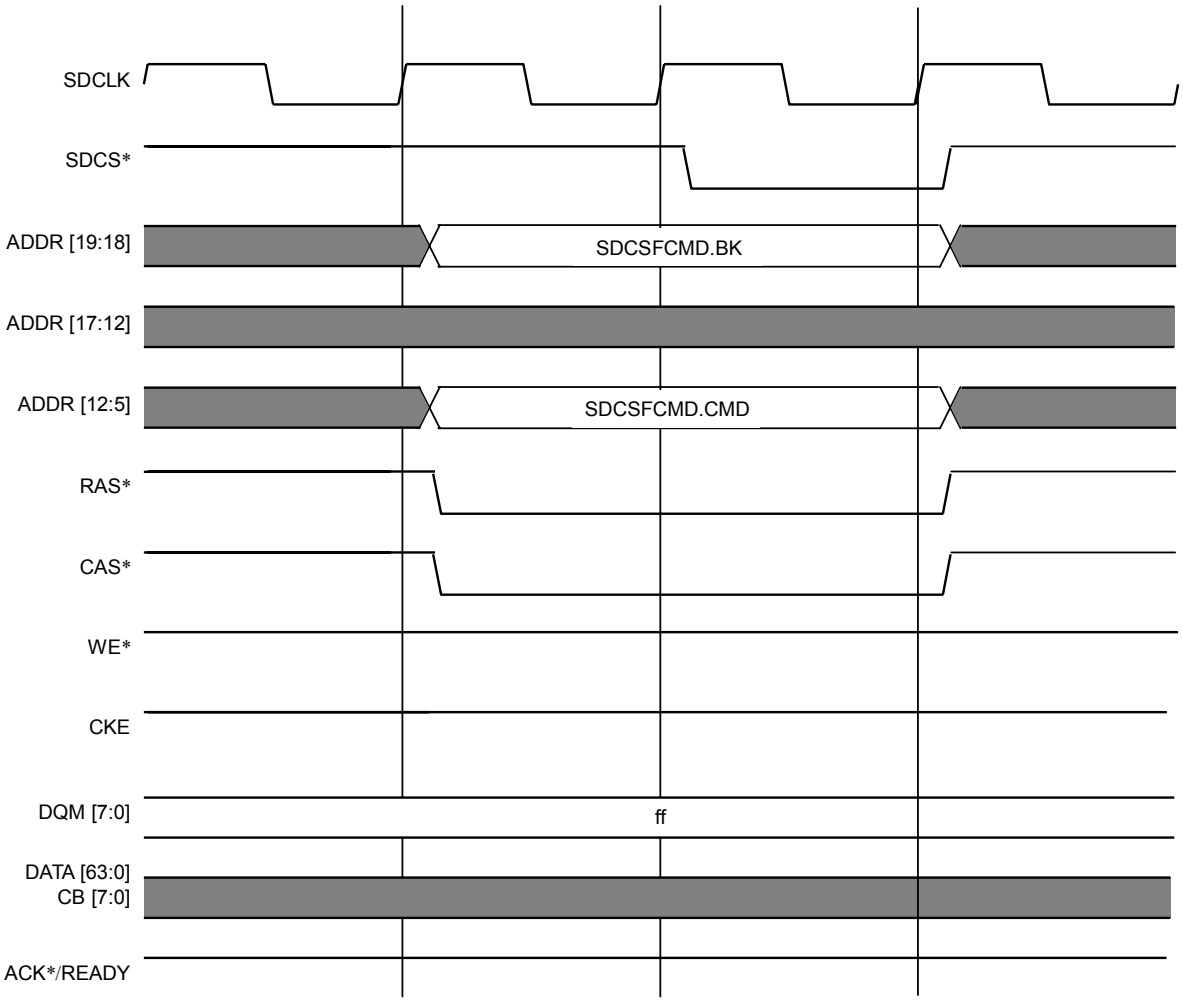


Figure 9.5.16 LCR Command

9.6 SDRAM Usage Example

Figure 9.6.1 illustrates an example SDRAM connection. Figure 9.6.2 illustrates an example SDRAM DIMM (168-pin) connection.

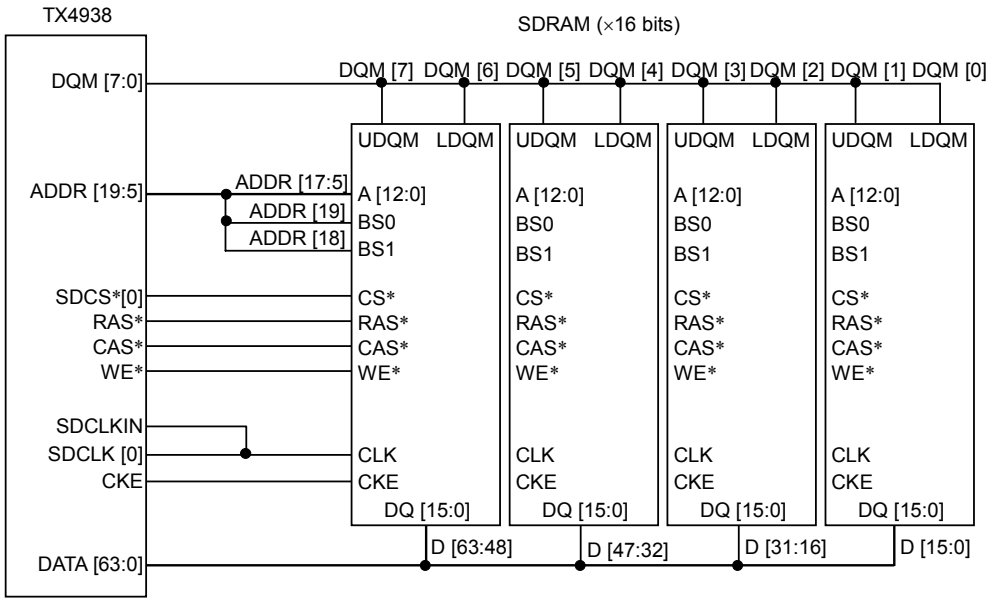


Figure 9.6.1 SDRAM (×16 bits) Connection Example

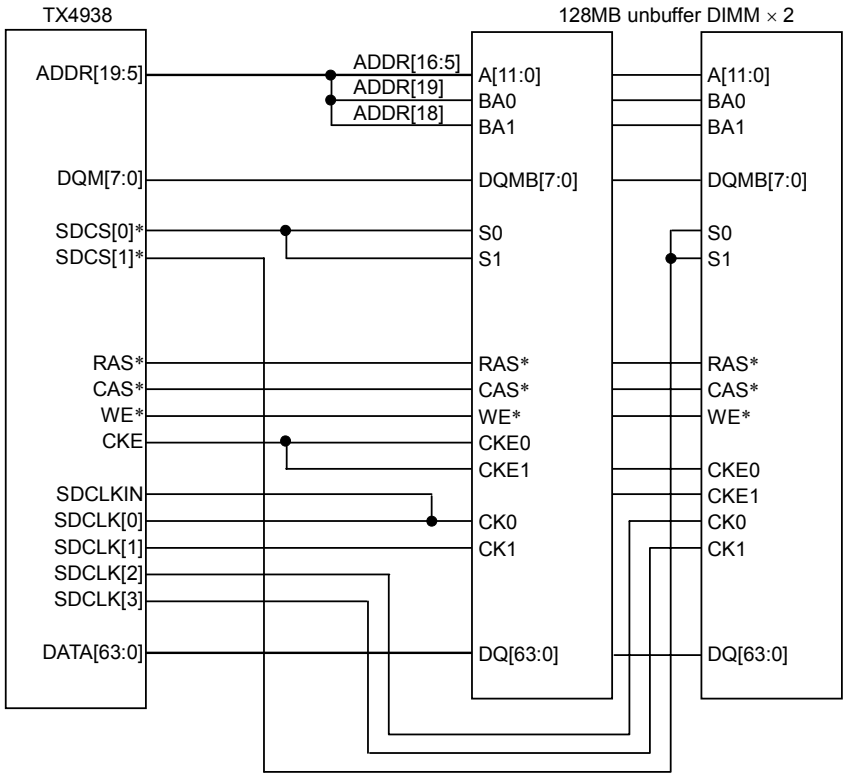


Figure 9.6.2 168-pin DIMM Connection Example

10. PCI Controller

10.1 Features

The TX4938 PCI Controller functions as a bus bridge between the TX4938 External PCI and the internal bus (G-Bus).

10.1.1 Overall

- Compliant to "PCI Local Bus Specification Revision 2.2"
- PCI Bus: 32-bit data bus; Internal Bus: 64-bit data bus
- Maximum PCI bus clock operating frequency: 66 MHz
- Dual address cycle support (40-bit PCI address space)
- Supports both the Initiator and Target functions
- Supports power management functions that are compliant to PCI Bus Power Management Interface Specifications Version 1.1.
- On-chip PCI Bus Arbiter, can connect to a maximum of four external bus masters
- 1-channel on-chip DMA Controller (PDMAC) dedicated to the PCI Controller
- Supports six PCI clock outputs
- The Internal Bus clock and PCI Bus clock are asynchronous and can be set independently
- Includes function for booting the TX4938 from memory on the PCI Bus
- Can set configuration data from serial ROM
- Mounted a retry function on the Internal Bus side also in order to avoid deadlock on the PCI Bus.

10.1.2 Initiator Function

- Single and Burst transfer from the Internal Bus to the PCI Bus
- Supports memory, I/O, configuration, special cycle, and interrupt acknowledge transactions.
- Address mapping between the Internal Bus and the PCI Bus can be modified
- Mounted 8-stage 64-bit data one FIFO each for Read and Write
- Post Write function enables quick termination of a maximum of four Write transactions by the G-Bus without waiting for completion on the PCI Bus.
- Endian switching function

10.1.3 Target Function

- Single and Burst transfer from the PCI Bus to the Internal Bus
- Supports memory, I/O, and configuration cycles
- Supports high-speed back-to-back transactions on the PCI Bus
- Address mapping between the PCI Bus and the Internal bus can be modified
- Mounted 8-stage 64-bit data FIFO for Read
- Mounted 12-stage 64-bit data FIFO for Write
- Post Write function enables quick termination of a maximum of nine Write transactions by the PCI Bus without waiting for completion on the G-Bus.
- Read Burst length (pre-fetch data size) on the Internal Bus when reading a pre-fetchable space can be made programmable
- Endian switching function

10.1.4 PCI Arbiter

- Supports four external PCI bus masters
- Uses the Programmable Fairness algorithm (two levels with different priorities for four round-robin request/grant pairs)
- Supports bus parking
- Bus master uses the Most Recently Used algorithm
- Unused slots and broken masters can be automatically disabled after Power On reset
- On-chip arbitration function can be disabled and external arbiter can be used

10.1.5 PDMAC (PCI DMA Controller)

- Direct Memory Access (DMA) Controller dedicated to 1-channel PCI
- Is possible to transfer data using minimal G-Bus bandwidth
- Data can be transferred bidirectionally between the G-Bus and the PCI Bus
- Specifying a physical address on the PCI Bus and an address on the G-Bus makes it possible to automatically transfer data between the PCI Bus and the G-Bus
- Supports the Chain DMA mode, in which a Descriptor containing chain-shaped addresses and a transfer size is automatically read from memory while DMA transfer continuous
- On-chip 4-stage 64-bit data buffer

10.2 Block Diagram

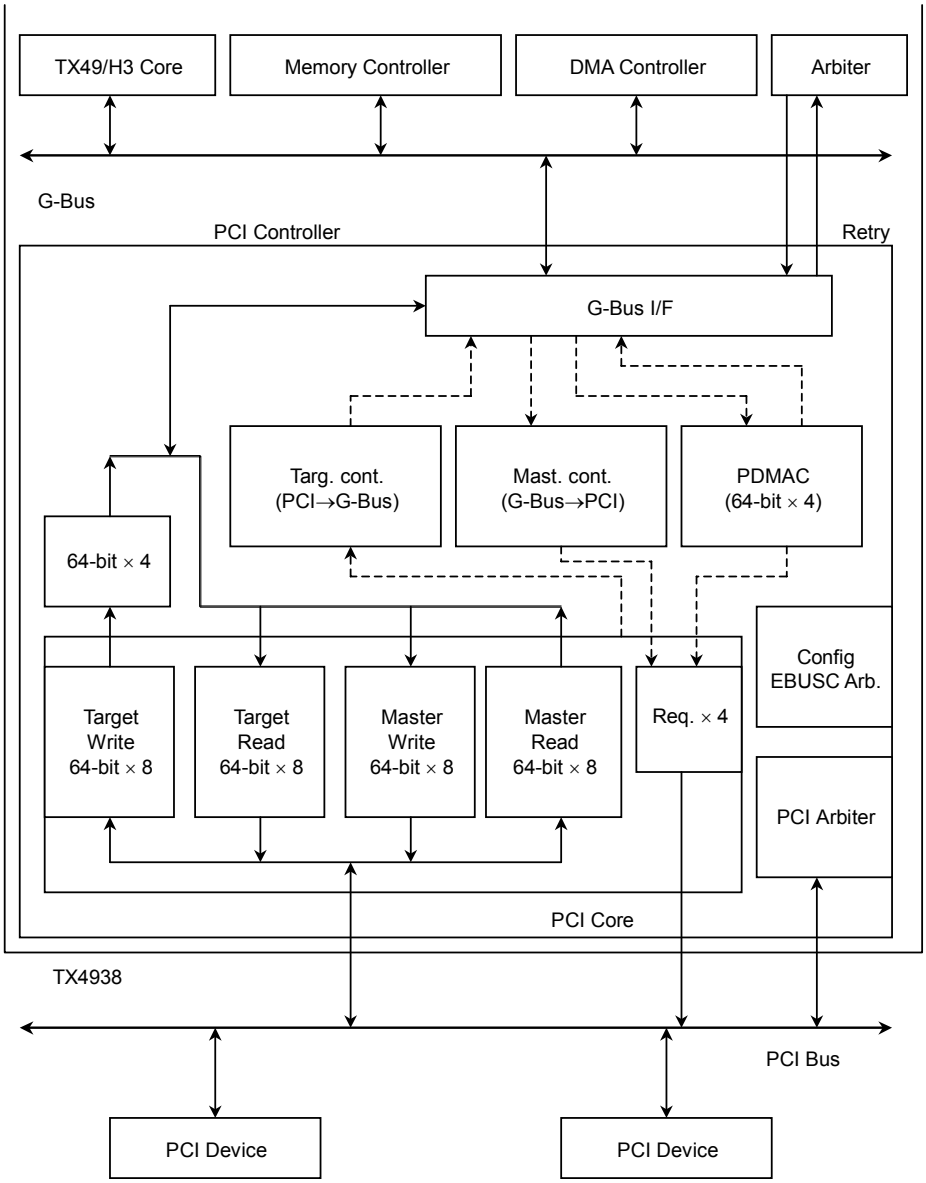


Figure 10.2.1 PCI Controller Block Diagram

10.3 Detailed Explanation

10.3.1 Terminology Explanation

The following terms are used in this chapter.

- Initiator
  Means the bus Master of the PCI Bus. The TX4938 operates as the initiator when it obtains the PCI Bus and issues PCI access.

- Target
  Means the bus Slave of the PCI Bus. The TX4938 operates as the target when an external PCI device on the PCI Bus executes PCI access to the TX4938.

- Host mode
  One PCI Host device exists for one PCI Bus. The PCI Host device uses a PCI configuration space to perform PCI configuration on other PCI devices on the PCI Bus.

  The TX4938 is set to the Host mode if the ADDR[19] signal is High when the RESET* signal is being deasserted.

- Satellite mode
  A PCI device other than the PCI Host device accepts configuration from the PCI Host device. This state is referred to as the Satellite mode.

  The TX4938 is set to the Satellite mode if the ADDR[19] signal is Low when the RESET signal is being deasserted.

- DWORD, QWORD
  DWORD expresses 32-bit words, and QWORD expresses 64-bit words. According to conventions observed regarding MIPS architecture, this manual uses the following expressions:
  Byte: 8-bit
  Half-word: 16-bit
  Word: 32-bit
  Double-word: 64-bit

10.3.2 On-chip Register

The PCI Controller on-chip register contains the PCI Configuration Space Register and the PCI Controller Control Register. The registers that can be accessed vary according to whether the current mode is the Host mode or the Satellite mode.

An external PCI Host device only accesses the PCI Configuration Space Register when in the Satellite mode. This register is defined in the PCI Bus Specifications. A PCI configuration cycle is used to access this register. This register cannot be accessed when in the Host mode. Section “10.5 PCI Configuration Space Register” explains each register in detail.

The PCI Controller Control Register is only accessed by the TX49 core and cannot be accessed from the PCI Bus.

Registers in the PCI Controller Control Register that include an offset address in the range from 0xD000 to 0xD07F can only be accessed when in the Host mode and cannot be accessed when in the Satellite mode. These registers correspond to PCI Configuration Space Registers that an external PCI Host device accesses when in the Satellite mode. Section "10.4 PCI Controller Control Register" explains each register in detail.

Figure 10.3.1 illustrates the register map when in the Host mode. Figure 10.3.2 illustrates the register map when in the Satellite mode.

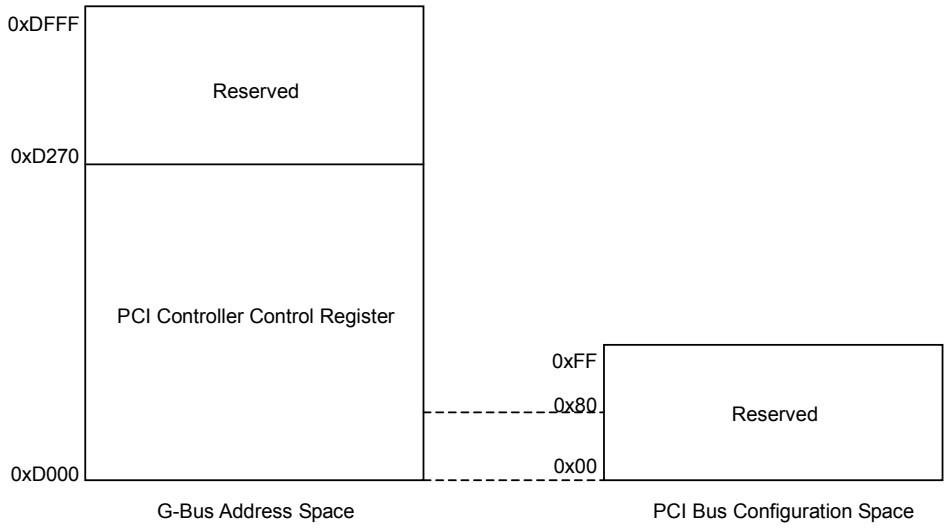


Figure 10.3.1 Register Map in the Host Mode

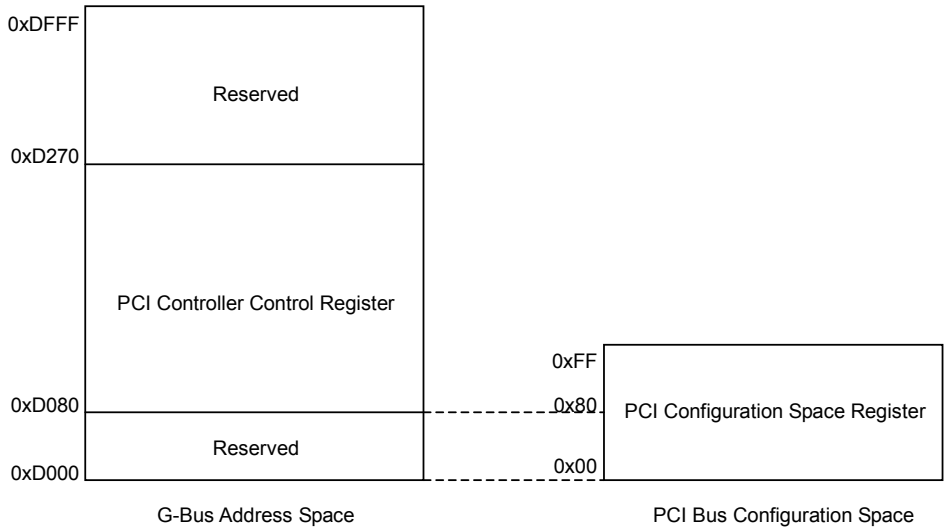


Figure 10.3.2 Register Map in the Satellite Mode

10.3.3 Supported PCI Bus Commands

Table 10.3.1 shows the PCI Bus commands that the PCI Controller supports.

Table 10.3.1 Supported PCI Bus Commands

C/BE Value	PCI Command	As Initiator	As Target
0000	Interrupt Acknowledge	†	—
0001	Special Cycle	†	—
0010	I/O Read	√	√
0011	I/O Write	√	√
0100	(Reserved)	—	—
0101	(Reserved)	—	—
0110	Memory Read	√	√
0111	Memory Write	√	√
1000	(Reserved)	—	—
1001	(Reserved)	—	—
1010	Configuration Read	†	‡
1011	Configuration Write	†	‡
1100	Memory Read Multiple	√	√
1101	Dual Address Cycle	√	√
1110	Memory Read Line	√	√
1111	Memory Write and Invalidate	√	√

Note: The byte enable signals are asserted as necessary during memory read and memory write cycles using I/O Read, I/O Write and Single Access commands. During burst memory reads, four byte enable signals are asserted.

Key: √ : Supported when in both the Host mode and the Satellite mode
† : Supported only when in the Host mode
‡ : Supported only when in the Satellite mode
— : Not supported

- I/O Read, I/O Write, Memory Read, Memory Write

  This command executes Read/Write access to the address mapped on the G-Bus and PCI Bus.

- Memory Read Multiple, Memory Read Line

  The Memory Read Multiple command is issued if all of the following conditions are met when the Initiator function is operating and Burst Read access is issued from the G-Bus to the PCI Bus.

  (1) A value other than “0” is set to the Cache Line Size Field (PCICFG1.CLS) of the PCI Configuration 1 Register.

  (2) The Read data word count is equal to or less than the value set in the Cache Line Size Field.

  Also, the Read Memory Line command is issued when all of the following conditions are met.

  (1) A value other than “0” is set to the Cache Line Size Field (PCICFG1.CLS) of the PCI Configuration 1 Register.

  (2) The Read data word count is equal to or less than the value set in the Cache Line Size Field.

The Memory Read command is issued if these conditions are not met, namely, if "0" is set to the Cache Line Size field (PCICFG1.CLS) of the PCI Configuration 1 Register. In the case of the target, a normal G-Bus cycle is issued to the address mapped from the PCI Bus to the G-Bus.

- Memory Write and Invalidate

When the TX4938 operates as the initiator, the PCI Controller is sue the Memory Write and Invalidate command if all of the folllowing conditions are met when write access from the G-Bus to the PCI Bus occurs.

(1) The Memory Write and Invalidate Enable bit (PCISTATUS.MWIEN) of the PCI Status Command Register is set.

(2) A value other than "0" was set to the Cache Line Size field (PCICFG1.CLS) of the PCI Configuration 1 Register.

(3) The word count of the Write data is equal to or larger than the value set in the Cache Line Size field.

The Memory Write command is issued in these conditions are not met.

When the TX4938 operates as the target, the Memory Write and Invalidate command is converted into G-Bus Write access. Note that the TX4938 does nto support the cache memory Snoop function.

- Dual address cycle

When the TX4938 operates as the initiator, the PCI Controller executes dual access cycles if the PCI Bus address exceeds 0x00_FFFF_FFFF.

When the TX4938 operates as the target, normal G-Bus cycles are executed to the address mapped from the PCI Bus to the G-Bus.

- Configuration Read, Configuration Write

These commands only issue configuration cycles as the when in the Host mode.

The corresponding configuration cycles are issued on the PCI Bus. This is done by either reading or writing from/to the G2P Configuration Data Register (G2PCFGDATA) after writing the configuration space address to the G2P Configuration Address Register. The TX4938 supports both "Type 0" and "Type 1" configuration transactions.

On systems that have PCI card slots, the PCI Host device checks each PCI card slot during system initialization to see if PCI device exist, then set the Configuration Space Register of the devices that do exist. If a PCI Configuration Read operation is performed for devices that do not exist, then by default a Bus Error exception will be generated since there is no PCI Bus response. Clearing the Bus Error Response During Initiator Read bit (PCICFG.IRBER) of the PCI Controller Configuration Register makes it possible to execute a Read transaction without causing a Bus Error. All bits of the data read at this time will be set to "1".

Configuration cycles will be accepted as the target only when in the Satellite mode. After reset, Retry response to PCI Configuration access will continue until the software sets the Target Configuration Access Ready Bit (PCICFG.TCAR) of the PCI Controller Configuration Register. Please use the software to set this bit after the software initialization process ends and the software is ready to accept PCI configuration.

- Interrupt Acknowledge

  This command issues interrupt acknowledge cycles as an initiator only when in the Host mode. Interrupt acknowledge cycles are executed on the PCI Bus when the G2P Interrupt Acknowledge Data Register (G2PINTACK) is read. The value returned by this Read becomes the interrupt acknowledge cycle data.

  The TX4938 does not support interrupt acknowledge cycles as the target.

- Special Cycle

  This command issues specialy cycles as the initiator only when in the Host mode. This command issues special cycles on the PCI Bus when writing to the G2P Special Cycle Data Register (G2PSPC). The written value is output as the special cycle data.

  The TX4938 does not support special cycles as the target.

10.3.4 Initiator Access (G-Bus → PCI Bus Address Conversion)

During PCI initiator access, the G-Bus address of the Burst transaction issued by the G-Bus that was converted into the PCI Bus address is used to issue a Burst transaction on the PCI Bus. 36-bit physical address (G-Bus addresses) are used on the G-Bus. Also, 40-bit PCI Bus addresses are used on the PCI Bus.

Three memory access windows and one I/O access window can be set in the G-Bus space (Figure 10.3.3). The size of each window is variable. When Burst transactions are issued to these access windows on the G-Bus, then that G-Bus address is converted into a PCI Bus address that is used to issue a Burst transaction to the PCI Bus as the initiator. PCI memory access is issued when the access window is the memory access window. PCI I/O access is issued when the access window is the I/O access window. Dual access cycles are also issued to the PCI Bus when the PCI Bus address exceeds 0x00_FFFF_FFFF.

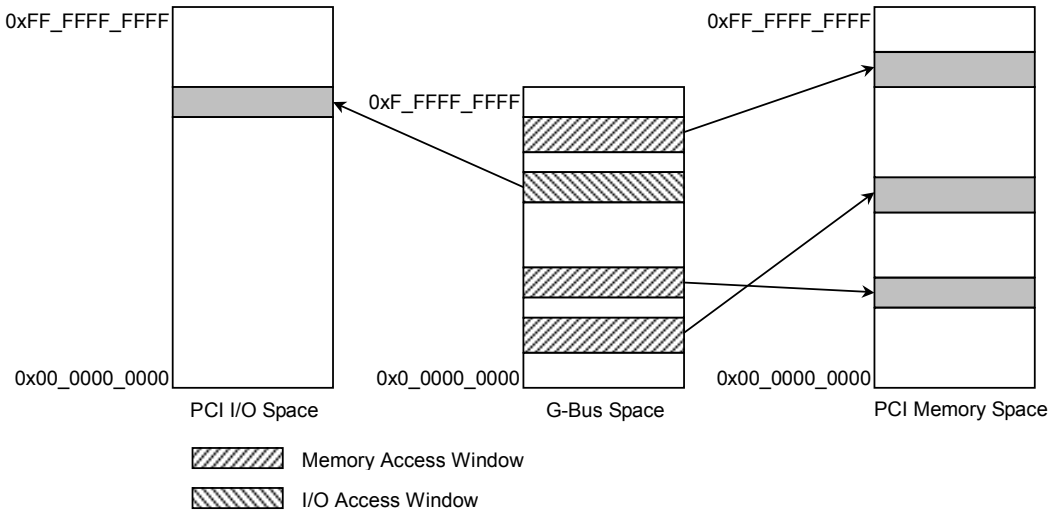


Figure 10.3.3 Initiator Access Memory Window

When expressed as a formula, conversion of a G-Bus address (GBusAddr[35:0]) into a PCI Bus Address (PCIAddr[39:0]) is as follows below. GBASE[35:8], PBASE[39:8], and AM[35:8] each represent the setting register of the corresponding access window indicated below in Table 10.3.2. The "&" symbol indicates a logical AND for each bit, "||" indicates a logical OR for each bit, "!" indicates logical NOT, and "|" indicates bit linking.

```
If (GBusAddr[35:8] & ! AM[35:8] == GBASE[35:8] & ! AM[35:8]) then
   PCIAddr[39:0] = PBASE[39:36]
| ((PBASE[35:8] & ! AM[35:8]) || (GBusAddr[35:8] & AM[35:8]))
| GBusAddr[7:0];
```

Table 10.3.2 Initiator Access Space Address Mapping Register

	G-Bus Base Address GBASE[35:8]	PCI Bus Base Address PBASE[39:8]	Address Mask AM[35:8]
Memory Space 0	G2PM0GBASE.BA[35:8]	G2PM0PBASE.BA[39:8]	G2PM0MASK.AM[35:8]
Memory Space 1	G2PM1GBASE.BA[35:8]	G2PM1PBASE.BA[39:8]	G2PM1MASK.AM[35:8]
Memory Space 2	G2PM2GBASE.BA[35:8]	G2PM2PBASE.BA[39:8]	G2PM2MASK.AM[35:8]
I/O Space	G2PIOGBASE.BA[35:8]	G2PIOPBASE.BA[39:8]	G2PIOMASK.AM[35:8]

Figure 10.3.4 illustrates this address conversion.

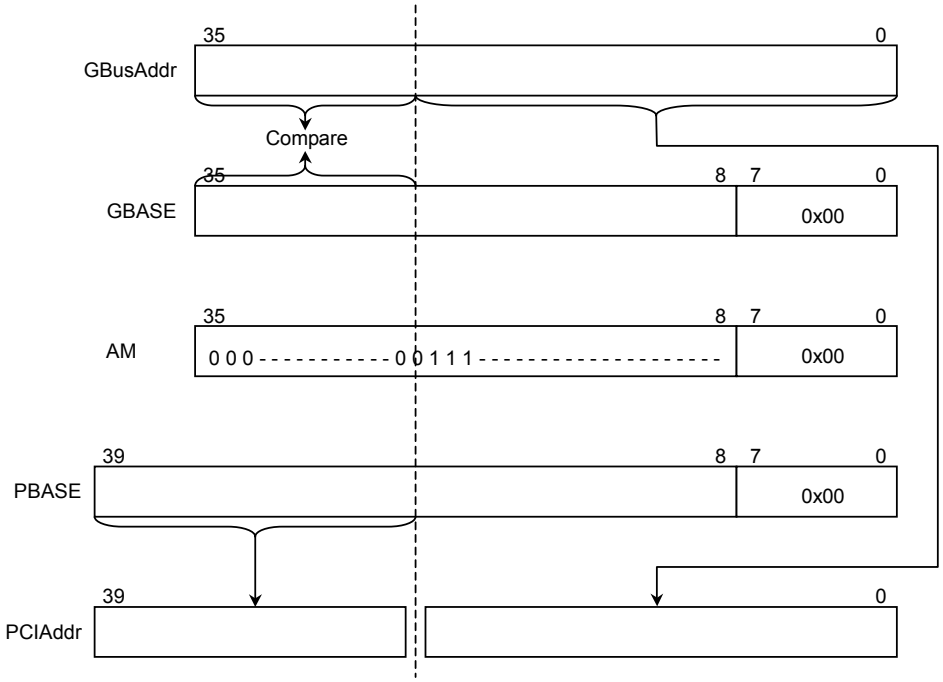


Figure 10.3.4 Address Conversion For Initiator (G-Bus → PCI Bus Address Conversion)

It is possible to set each space to valid/invalid or to perform Word Swap (see "10.3.7 Endian Switching Function). Table 10.3.3 shows the settings registers for these properties.

When 64-bit access is made to the initiator memory space, two 32-bit Burst accesses are issued on the PCI Bus. 64-bit access to the I/O space is not supported.

Also, operation is not guaranteed if resources in the PCI space were made cacheable and were then accessed when the Critical Word First function of the TX49/H3 core was enabled.

Table 10.3.3 Initiator Access Space Properties Register

	Enable	Word Swap
Memory Space 0	*BusMasterEnable* & PCICCFG.G2PM0EN	G2PM0GBASE.BSWAP
Memory Space 1	*BusMasterEnable* & PCICCFG.G2PM1EN	G2PM1GBASE.BSWAP
Memory Space 2	*BusMasterEnable* & PCICCFG.G2PM2EN	G2PM2GBASE.BSWAP
I/O Space	*BusMasterEnable* & PCICCFG.G2PIOEN	G2PIOGBASE.BSWAP

BusMasterEnable:
Host mode: PCI State Command Register Bus Master Bit (PCISTATUS.BM)
Satellite mode: Command Register Bus Master bit

10.3.5 Target Access (PCI Bus → G-Bus Address Conversion)

During PCI target access, the PCI Bus address of the Bus transaction issued by the PCI Bus is converted into a G-Bus address and is used to issue a Bus transaction on the G-Bus. 40-bit PCI Bus addresses are used on the PCI Bus. Also, 36-bit physical addresses are used on the G-Bus.

Three memory access windows and one I/O access window can be set in the PCI bus space (Figure 10.3.5). The size of each window is fixed. When Bus transactions to these access windows is issued on the PCI Bus, these Bus transactions are accepted as PCI target devices. The PCI Bus Address is converted into G-Bus addresses, then Bus transactions are issued to the G-Bus.

The memory space window responds to the PCI memory space access command. The I/O space window responds to the PCI I/O space access command.

Note: Byte swapping is always disabled when prefetch mode is disabled. When the G-Bus is configured for big-endian mode, the order of bits in a 32-bit word does not change during a PCI transfer. (The byte ordering changes.)

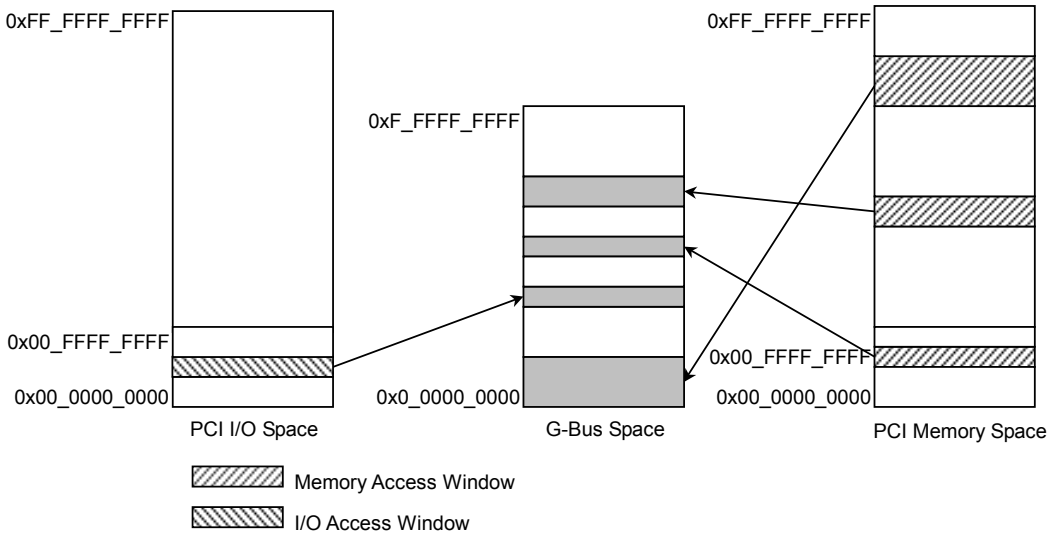


Figure 10.3.5 Target Access Memory Window

When expressed as a formula, conversion of a PCI Bus Address (PCIAddr[39:0]) into a G-Bus address (GBusAddr[35:0]) is as follows below. GBASE[35:8], and PBASE[39:8] each represent the setting register of the corresponding access window indicated below in Table 10.3.4. The "&" symbol indicates a logical AND for each bit, and "|" indicates bit linking.

Memory space 0

```
If (PCIAddr[39:29]  == P2GM0PUBASE.BA[39:32] | P2GM0PLBASE.BA[31:29] then
   GBusAddr[35:0] = P2GM0GBASE[35:29] | PCIAddr[28:0];
```

Memory space 1

```
If (PCIAddr[39:24]  == P2GM1PUBASE.BA[39:32] | P2GM1PLBASE.BA[31:24] then
   GBusAddr[35:0] = P2GM1GBASE[35:24] | PCIAddr[23:0];
```

Memory space 2

```
If (PCIAddr[31:20]  == P2GM2PBASE.BA[31:20]) then
   GBusAddr[35:0] = P2GM2GBASE[35:20] | PCIAddr[19:0];
```

I/O space

```
If (PCIAddr[31:8]  == P2GIOPBASE.BA[31:8]) then
   GBusAddr[35:0] = P2GIOGBASE[35:8] | PCIAddr[7:0];
```

Table 10.3.4 Target Access Space Address Mapping Register

	Space Size	PCI Address	PCI Bus Base Address PBASE	G-Bus Base Address GBASE
Memory Space 0	512 MB	40-bit	P2GM0PUBASE.BA[39:32] | P2GM0PLBASE.BA[31:29]	P2GM0GBASE.BA[35:29]
Memory Space 1	16 MB	40-bit	P2GM1PUBASE.BA[39:32] | P2GM1PLBASE.BA[31:24]	P2GM1GBASE.BA[35:24]
Memory Space 2	1 MB	32-bit	P2GM2PBASE.BA[31:20]	P2GM2GBASE.BA[35:20]
I/O Space	256 B	32-bit	P2GIOPBASE.BA[31:8]	P2GIOGBASE.BA[35:8]

Figure 10.3.6 illustrates this address conversion.

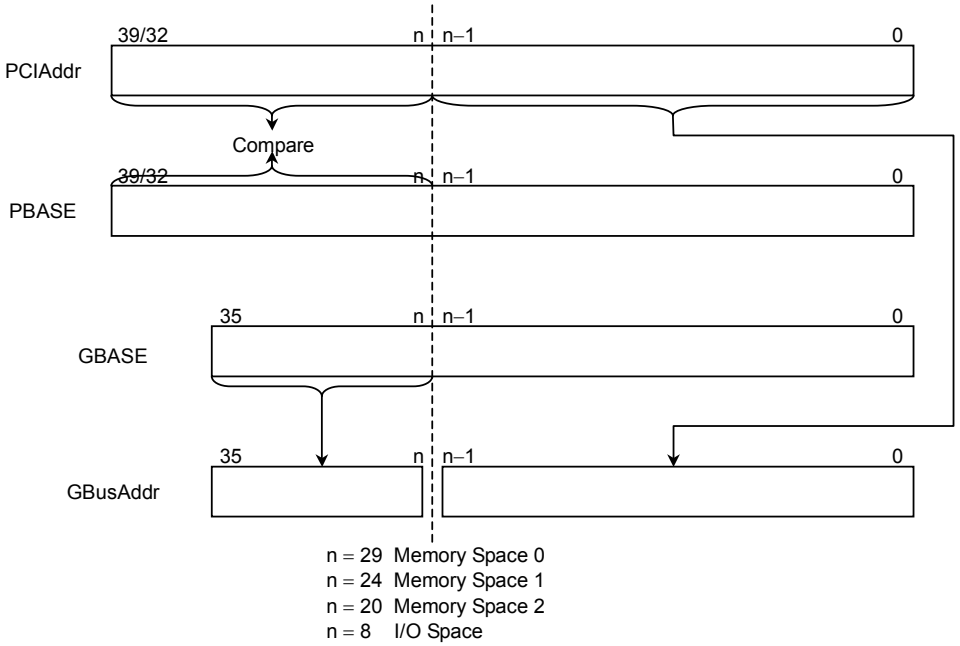


Figure 10.3.6 Address Conversion for Target (PCI Bus (PCI Bus → G-Bus Address Conversion)

It is possible to set each space to valid/invalid, pre-fetch Read to valid/invalid, or to perform Word Swap (see10.3.7). Table 10.3.5 shows the settings registers for these properties.

When pre-fetch Reads are set to valid, data transfer is performed on the G-Bus according to the size set by the Target Pre-fetch Read Burst Length Field (P2GCFG.TPRBL) of the P2G Configuration Register during a PCI target Read transaction. This is performed using accesses to resources that will not be affected even if a pre-read such as memory is performed. Also, PCI Burst Reads to memory spaces that were set to I/O space and pre-fetch disable are not supported.

Note: Always use PCI single reads. Don't use burst reads.

Table 10.3.5 Target Access Space Properties Register

	Enable	Pre-fetch (Initial State)	Word Swap
Memory Space 0	PCICCFG.TCAR & *MemEnable* & P2GM0GBASE.P2GM0EN	P2GCFG.MEM0PD (valid)	P2GM0GBASE.BSWAP
Memory Space 1	PCICCFG.TCAR & *MemEnable* & P2GM1GBASE.P2GM1EN	P2GCFG.MEM1PD (valid)	P2GM1GBASE.BSWAP
Memory Space 2	PCICCFG.TCAR & *MemEnable* & P2GM2GBASE.P2GM2EN	P2GCFG.MEM2PD (invalid)	P2GM2GBASE.BSWAP
I/O Space	PCICCFG.TCAR & *IOEnable* & P2GIOGBASE.P2GIOEN	Always invalid	P2GIOGBASE.BSWAP

MemEnable:
Host mode: PCI State Command Register Memory Space bit (PCISTATUS.MEMSP)
Satellite mode: Command Register Memory Space bit

IOEnable:
Host mode: PCI State Command Register I/O Space bit (PCISTATUS.IOSP)
Satellite mode: Command Register I/O Space bit

10.3.6 Post Write Function

The Post Write function improves system performance by completing the original bus Write transaction without waiting for the other bus to complete its transaction when the first bus issues a Write transaction. Initiator Write can Post Write a maximum of four Write transactions, and Target Write can Post Write a maximum of nine Write transactions.

Due to compatibility issues with old PC software in the PCI specifications, performing Post Writes with Initiator Configuration Write and Target I/O Write is not recognized. However, the TX4938 PCI Controller can even perform Post Writes to these functions. In order to guarantee that these Writes are completed by the target device, please execute Reads to the device that performed the Write, then either refer to the read value (so the TX49/H3 core can support non-blocking load) or execute the SYNC instruction.

10.3.7 Endian Switching Function

The TX4938 supports both the Little Endian mode and the Bit Endian mode. On the other hand, the PCI Bus is only defined in Little Endian logic. Therefore, when the TX4938 is in the Big Endian mode, either the software or the hardware must perform some kind of conversion when exchanging data larger than 2 B in size with the PCI Bus.

The PCI Controller can specify the endian switching function that reverses the byte arrangement of the DWORD (32-bit) data for each access window.

Initial state operation matches the correspondence between the address and byte data regardless of the endian mode (operation is address consistent). For example, if WORD (16-bit) data is written to address 0 of the PCI Bus when the TX4938 is in the Big Endian mode, the upper byte (address 0 in Big Endian) is written to PCI Bus address 0 and the lower byte (address 1 in Big Endian) is written to address 1 of the PCI Bus. For Little Endian PCI devices, this means that the byte order is reversed.

When in the Big Endian mode and a particular access window Endian switching mechanism is validated, data is transferred so the byte order does not change in DWORD (32-bit) access to that access window.

Endian switching during initiator access is specified by the Byte Swap bit (BSWAP) of the G-Bus Base Address Register (G2PMnGBASE, G2PIOGBASE) of the access window for each initiator access (see Table 10.3.3).

Ending switching during target access is specified by the Byte Swap bit (BSWAP) of the G-Bus Base Address Register (P2GMnGBASE, P2GIOGBASE) of the access window for each target access (see Table 10.3.5).

10.3.8 66 MHz Operation Mode

The TX4938 PCI Controller supports 66 MHz PCI. When in the Host mode, the procedure for setting the PCI Bus to the 66 MHz mode is as follows below.

(1) Start the system with a PCI Bus Clock frequency of 33 MHz or less.

(2) The TX4938 system initialization program checks the 66 MHz Capable bit (bit 5) of the configuration Space Register Status Register in all PCI devices. If the 66 MHz Capable bit of all devices is set, then change the PCI Bus Clock frequency according to the following procedure.

(3) Assert the PCI Bus Reset signal. (The TX4938 does not have PCI Reset output, so it is necessary to use an external circuit to control the PCI Bus Reset signal.)

(4) Set the Software Reset bit (PCICFG.SRST) of the PCI Controller Configuration Register.

(5) Setting the PCI66 MHz Mode bit (CCFG.PCI66) of the Chip Configuration Register asserts the M66EN signal.

(6) Modifying the setting of the PCICLK Division Ratio field (CCFG.PCIDIVMODE) of the Chip Configuration Register changes the PCI Clock frequency from 33 MHz to 66 MHz.

(7) The software reset bit (PCICCFG.SRST) is cleared after the PLL stabilizes (about 10 ms).

(8) Deassert the PCI Bus Reset signal. Each PCI device detects assertion of the M66EN signal if necessary and performs the process.

When the TX4938 is in the Satellite mode, the M66EN signal becomes the input signal. It is possible to read this state from the 66 MHz Drive Status bit (P2GSTATUS.M66EN) of the P2G Status Register.

PCI Reset is detected by either using the PCI Bus Reset Signal as the TX4938 overall reset signal or using the PCI Bus Reset Signal assertion detection device that the system provides. Then, the software reset the PCI Controller. The software uses a hardware reset (PCICCFG.HRST) of the PCI Controller Configuration Register to reset the PCI Controller.

10.3.9 Power Management

The TX4938 PCI Controller supports power management functions that are compliant to PCI Bus Power Management Interface Specifications Version 1.1.

The PCI Host device controls the system status by reporting the power management state to the PCI Satellite device. Also, the PCI Satellite device uses the PME* signal to report requests for changing the power management state or to report to the PCI Host device that a power management event has occurred.

10.3.9.1 Power Management State

In the case of the PCI Bus Power Management Interface Specifications, four power management states are defined from State D0 to State D3. The TX4938 supports states D0 through D3. Figure 10.3.7 illustrates the power management state transition.

After Power On Reset, or when transitioning from the $D3_{HOT}$ state to the D0 state, the power management state becomes uninitialized D0. If initialized by the system software at this point, the state transitions to D0 Active.

If an external PCI Host device writes 11b ($D3_{HOT}$) to the PowerState field of the Power Management Control Status Register (PMCSR) of the Configuration space when in the Satellite mode, then the Power Management State Change bit (P2GSTATUS.PMSC) of the P2G Status Register is set and transitions to the $D3_{HOT}$ state. It then becomes possible to report Power State Change interrupts. The PowerState field value can be read from the PowerState field (PCISSTATUS.PS) of the Satellite Mode PCI Status Register.

The TX4938 uses the software to change the system status after a status change is detected.

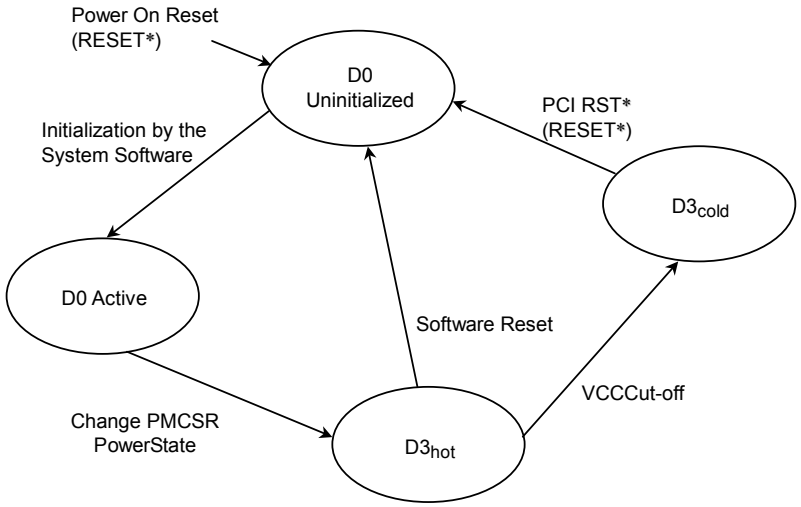


Figure 10.3.7 Transition of the Power Management States

10.3.9.2 PME* Signal (Satellite Mode)

The following PMEs (Power Management Events) are reported when in the Satellite mode.

- The PCI Host device sets the PME_En bit of the PMCSR Register in the TX4938 Configuration space. This makes it possible for the TX4938 to assert the PME* signal.

  Then, the PME_En Set bit (P2GSTATUS.PMEES) of the P2G Status Register is set. Furthermore, it also becomes possible to generate PME_En Set interrupts. The PME_En bit value can be read from the PME_En bit (PCISSTATUS.PMEEN) of the Satellite Mode PCI Status Register.

- Writing "1" to the PME bit (P2GCFG.PME) of the P2G Configuration Register sets the PME_Status bit of the PMCSR Register, then asserts the PME* signal, which is the open drain signal. PME is then reported to the PCI Host device.

- The PCI Host device checks the PMCSR PME_Status bit of each PCI device, then specifies the PCI device that asserted the PME* signal.

- After the process corresponding to PME ends, the PCI Host device writes "1" to the TX4938 PME_Status bit that reported PME, thereby reporting the end of the process. As a result, the PME_Status bit of the PMCSR Register is cleared and the PME* signal is deasserted.

  Then, the PME Status Clear bit (P2GSTATUS.PMECLR) of the P2G Status Register is set. It is also possible to generate PME Status Clear interrupts.

10.3.9.3 PME* Signal* (Host Mode)

The PME Detection bit (PCICSTATUS.PMED) of the PCI Controller Status Register is set when an external satellite device asserts the PME* signal while the TX4938 is in the Host mode. It is also possible to generate PME Detection interrupts at this time.

10.3.10 PDMAC (PCI DMA Controller)

The PCI DMA Controller (PDMAC) is a one-channel PCI Director Memory Access (DMA) controller. Data can be transferred bidirectionally between the G-Bus and the PCI Bus.

Note: The PDMAC can only access the SDRAMC on the G-Bus. It does not provide support for access to other controllers on the G-Bus.

10.3.10.1 DMA Transfer

The following DMA transfer procedure does not use the Chain DMA mode.

1. Address Register and Count Register Setting
   Sets values for the three following registers.
   - PDMAC G-Bus Address Register (PDMGA)
   - PDMAC PCI Bus Address Register (PDMPA)
   - PDMAC Count Register (PDMCTR)

2. Chain Address Register Setting
   Sets "0" to the PDMAC Chain Address Register (PDMCA).

3. PDMAC Status Register (PDMSTATUS) Clearing
   Clears any remaining status from a previous DMA transfer.

4. PDMAC Configuration register (PDMCFG) Setting
   Clears the Channel Reset bit (CHRST), and makes settings such as the data transfer direction (XFRDIRC), and the data transfer unit size (XFRSIZE).

5. DMA Transfer Initiation
   Setting the Transfer Active bit (XFRACT) of the PDMAC Configuration Register initiates DMA transfer.

6. Termination Report
   When the DMA data transfer terminates normally, the Normal Data Transfer Complete bit (NTCMP) of the PDMAC Status Register (PDMSTATUS) is set. An interrupt is then reported if the Normal Data Transfer Complete Interrupt Enable bit (NTCMPIE) of the PDMAC Configuration Register is set.

   If an error is detected during DMA transfer, the error cause is recorded in the lower 5 bits of the PDMAC Status Register and the transfer is aborted. An interrupt is then reported if the Error Detection Interrupt Enable bit (ERRIE) of the PDMAC Configuration register is set.

10.3.10.2 Chain DMA

DMA Command Descriptors are 4 QWORD (32-Byte) data structures indicated in Table 10.3.6 that are placed in memory.

Storing the starting memory address of another DMA Command Descriptor in the Offset 0 Chain Address Field makes it possible to configure a chain list for the DMA command Descriptor. Set “0” in the Chain Address field of the DMA Command Descriptor at the end of the chain list.

When the DMA transfer specified by one DMA Command Descriptor ends, the PDMAC reads the next DMA Command Descriptor that the Chain Address field automatically points to, then continues the DMA transfer. Such continuous DMA transfer that uses multiple descriptors in a chain format is referred to as the Chain DMA mode.

When a DMA Command Descriptor is placed to an address that does not extend across a 32 QWORD boundary in memory, this transfer method is more efficient since data can be read by a single G-Bus Burst Read transaction.

Table 10.3.6 DMA Command Descriptors

Offset Address	Field Name	Transfer Destination Register
0x00	Chain Address	PDMAC Chain Address Register (PDMCA)
0x08	G-Bus Address	PDMAC G-Bus Address Register (PDMGA)
0x10	PCI Bus Address	PDMAC PCI Bus Address Register (PDMPA)
0x18	Count	PDMAC Count Register (PDMCTR)

The DMA transfer procedure is as follows when in the Chain DMA mode.

1. Count Register Setting
   Sets “0” to the PDMAC Count Register (PMDCTR).

2. DMA Command Descriptor Chain Construction
   Constructs the DMA Command Descriptor Chain in memory.

3. PDMAC Status Register (PDMSTATUS) Clearing
   Clears any remaining status from a previous DMA transfer.

4. PDMAC Configuration Register (PDMCFG) Setting
   Clears the Channel Regster bit (CHRST) and makes settings such as the data transfer direction (XFRDIRC) and the data transfer unit size (XFRSIZE).

5. DMA Transfer Initiation
   Setting the address of the DMA Command descriptor that is at the beginning of the Chain List in the PDMAC Chain Address Register (PDMCA) automatically initiates DMA transfer.

   First, the values stored in each field of the DMA Command Descriptor that is at the beginning of the Chain List are read to each corresponding PDMAC Register, then DMA transfer is performed according to the read values.

   If a value other than “0” is stored in the PDMAC Chain Address Register (PDMCA), data transfer of the size stored in the PDMAC Count Register is complete, then the DMA Command Descriptor value for the memory address specified by the PDMAC Chain Address Register is read.

   When the Chain Address field value reads a descriptor of “0”, the PDMAC Chain Address Register value is not updated and the previous value (address of the Data Command Descriptor at which the Chain Address field value is “0” when read) is held.

   0 value judgement is performed when the lower 32 bits of the PDMAC Chain Address Register are rewritten. DMA transfer is automatically initiated if the value was not “0”. Therefore, please write to the upper 32 bits first when writing to the PDMAC Chain Address Register using a 32-bit Store instruction.

6. Termination Report
   When DMA data transfer of all descriptor chains terminates normally, the Normal Chain Complete bit (NCCMP) of the PDMAC Status Register is set. An interrupt is reported if the Chain Termination Interrupt Enable bit (MCCMPIE) of the PDMAC Configuration register (PDMCFG) is set.

   Also, the Normal Data Transfer Complete bit (NTCMP) of the DPMAC Status Register is set each time the DMA data transfer specified by a DMA Command Descriptor terminates normally. An interrupt is reported if the Normal Data Transfer Complete Interrupt Enable bit (NTCMPIE) of the PDMAC Configuration Register (PDMCFG) is set.

   If an error is detected during DMA transfer, the error cause is recorded in the lower 5 bits of the PDMAC Status Register and the transfer is aborted. An interrupt is then reported if the Error Detection Interrupt Enable bit (ERRIE) of the PDMAC Configuration register is set.

10.3.10.3 Dynamic Chain Operation

It is possible to dynamically add other DMA Command Descriptor Chains to a DMA Command Descriptor Chain that is currently being processed when executing DMA data transfer. This is done according to the following procedure.

1. DMA Command Descriptor Chain Construction
   Constructs a DMA Command Descriptor Chain in memory.

2. Addition of DMA Command Descriptor Chains
   Substitutes the address of the command descriptor that is at the beginning of the descriptor chain to be added into the Descriptor Chain Address field at the end of the DMA Command Descriptor Chain that is currently performing DMA transfer.
3. Chain Enable bit checking
   Reads the value of the Chain Enable bit (CHNEN) in the PDMAC Configuration Register (PDMCFG). If the read value is "0", then the Chain Address field value of the DMA Command Descriptor indicated by the address stored in the PDMAC Chain Address Register (PDMCA) is written to the PDMAC Chain Address Register (PDMCA)

10.3.10.4 Data Transfer Size

The Transfer Size field (PDMCFG.XFERSIZE) of the PDMAC Configuration Register specifies the transfer size of each G-Bus transaction in a DMA transfer. The transfer size can be selected from one of the following: 1 DWORD, 1 QWORD, or 4 QWORD (Burst transfer).

1 QWORD or 4 QWORD can only be selected as the transfer size when the setting of the PDMAC G-Bus Address Register (PDMGA) and the PDMAC PCI Bus Address Register (PDMPA) is a 64-bit address boundary and the PDMAC Count Register (PDMCTR) setting is an 8-byte multiple. 1 DWORD must be selected as the transfer size in all other cases.

10.3.11 Error Detection, Interrupt Reporting

The PCI Controller reports the four following types of interrupts to the Interrupt Controller (IRC).

- Normal Operation Interrupt (Interrupt Number: 16, PCIC)
- PDMAC Interrupt (Interrupt Number: 15, PDMAC)
- Power Management Interrupt (Interrupt Number: 23, PCIPME)
- Error Detection Interrupt (Interrupt Number: 22, PCIERR)

When each cause is detected, an interrupt is reported if the corresponding Status bit is set, and the corresponding Interrupt Enable Bit is set. The following tables list the name of each interrupt cause, the Status bit, and the Interrupt Enable bit. Please refer to the explanation of each Status bit for more information regarding each interrupt cause.

10.3.11.1 Normal Operation Interrupt

Name	Status Bit		Interrupt Enable Bit	
M66EN Signal Assert Detect	P2GSTATUS	M66EN	P2GMASK	M66ENIE

10.3.11.2 PDMAC Interrupts

Name	Status Bit		Interrupt Enable Bit	
Normal Chain Termination	PDMSTATUS	NCCMP	PDMCFG	NCCMPIE
Normal Data Transfer Termination		NTCMP		NTCMPIE
Inter-Transfer Stall Time Reached		STLTRF		ERRIE
Configuration Error		CFGERR		
PCI Fatal Error		PCIERR		
G-Bus Chain Error		CHNERR		
G-Bus Data Error		DATAERR		

10.3.11.3 Power Management Interrupts

Name	Status Bit		Interrupt Enable Bit	
PM Status Change Detect	P2GSTATUS	PMSC	P2GMASK	PMSCIE
PME_En Set Detect		PMEES		PMEESIE
PME Status Clear Detect		PMECLR		PMECLRIE
PME Detect	PCICSTATUS	PME	PCICMASK	PMEIE

10.3.11.4 Error Detection Interrupts

Name	Status Bit		Interrupt Enable Bit	
Parity Error Detect	PCISTATUS / PCISSTATUS	DPE	PCIMASK	DPEIE
System Error Report		SSE		SSEIE
Master Abort Receive		RMA		RMAIE
Target Abort Receive		RTA		RTAIE
Target Abort Report		STA		STAIE
Master Data Parity Error		MDPE		MDPEIE
TRDY Timeout Error	G2PSTATUS	IDTTOE	G2PMASK	IDTTOEIE
Retry Timeout Error		IDRTOE		IDRTOEIE
Broken Master Detect	PBASTATUS	BMD	PBAMASK	BMDIE
Long Burst Transfer Detect	PCICSTATUS	TLB	PCICMASK	TLBIE
Negative Increase Burst Transfer Detect		NIB		NIBIE
Zero Increase Burst Transfer Detect		ZIB		ZIBIE
PERR* Detect		PERR		PERRIE
SERR* Detect		SERR		SERRIE
G-Bus Bus Error Detect		GBE		GBEIE

Note: In the initiator write cycle, access on the G-Bus has been finished before access on the PCI bus is finished (Post write). Therefore, when an error occurs on PCI bus, it is reported with an error detection interrupt, as shown above.
In the initiator read cycle, when an error occurs on the PCI bus access, PCIC responds with a G-Bus error instead of returning read data to the G-Bus. Setting "0" to the IRBER bit of the PCICFG register suppresses output of a G-Bus error during initiator read.

10.3.12 PCI Bus Arbiter

Configuration settings (DATA[2] signal) during boot up select whether to use the on-chip PCI Bus arbiter (Internal PCI Bus Arbiter mode) or to use the External PCI Bus arbiter (External PCI Bus Arbiter mode).

When in the Internal PCI Bus Abiter mode, setting the PCI Bus Arbiter Enable bit (PBACFG.PBAEN) of the PCI Bus Arbiter Configuration Register starts operation.

The on-chip PCI Bus arbiter can arbitrate eight sets of PCI Bus usage requests from the Bus Master. Five ports are used: one for the PCI Controller bus master and four for External Bus masters. The three remaining ports are reserved for future expanded features.

10.3.12.1 Request Signal, Grant Signal

The four external Bus Masters are connected to the REQ[3:0] signal and the GNT[3:0]* signal.

Also, when in the External PCI Bus Master mode, the REQ[0]* signal becomes the PCI Bus Request Output signal and the GNT[0]* signal becomes the Bus Usage Permission Input Signal. Furthermore, the REQ[1]* signal can be used as an interrupt output signal to the external devices (see 14.3.7 for more information).

10.3.12.2 Priority Control

As illustrated below in Figure 10.3.8, a combination of two round-robin sequences is used as the arbitration algorithm that determines the priority of Internal PCI Bus arbiter bus requests. The round-robin with the lower priority (Level 2) consists of Masters W - Z, and the round-robin with the high priority (Level 1), consists of Master A - D and Level 2 Masters. The PCI Bus Arbiter Request Port Register (PBAREQPORT) specifies whether to allocate the PCI Controller and the four External Bus Masters to Masters A-D or W - Z.

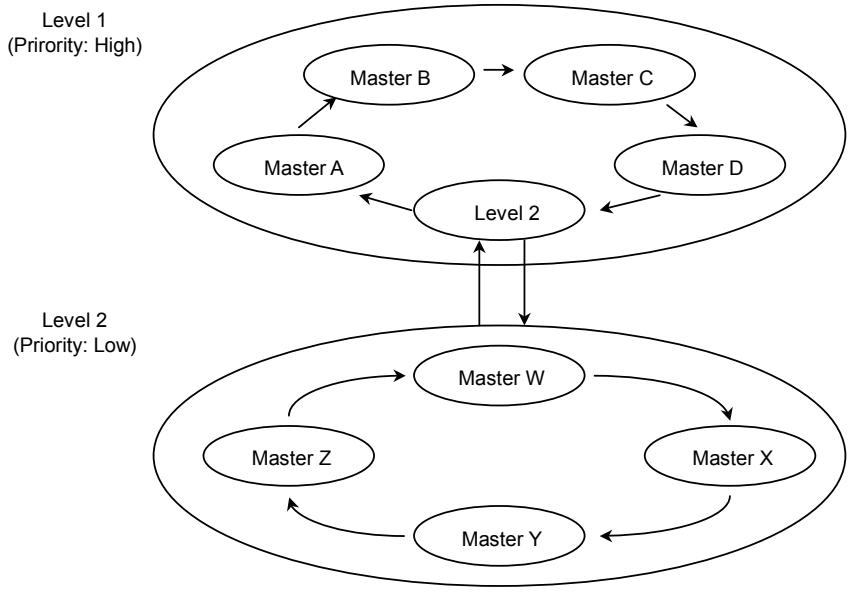


Figure 10.3.8 PCI Bus Arbitration Priority

The Bus Master priority is determined based on the Level 1 round-robin sequence. However, when Level 2 is used inside Level 1, the Level 2 Bus Master priority is determined based on the Level 2 round-robin sequence.

All 8 Bus Masters cannot be used on the TX4938. However, the Bus Master priority would be as follows if we assume there is a hypothetical device that can use all 8 Bus Masters and all 8 Bus Masters (Masters A – D, W – Z) simultaneously requested the bus.

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow W$
$\rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow X$
$\rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow Y$
$\rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow Z$
$\rightarrow A$ (returns to the beginning)

Since the priority can only transition in the order indicated by the above arrows (or the arrows in Figure 10.3.8, if we assume that the three Bus Masters A, B, and W exist, then Master B will obtain the bus first. If A and W then simultaneously request the bus, then PCI Bus ownership will transition in the order $B \rightarrow W \rightarrow A$.

10.3.12.3 Bus Parking

The On-chip PCI Bus Arbiter supports bus parking.

The last PCI Bus Master is made the Park Master when the Fix Park Master bit (FIXPM) of the PCI Bus Arbiter Configuration Register (PBACFG) is cleared (in the default state). When this bit is set, the Internal PCI Bus Arbiter Request A Port (Master A) becomes the Park Master.

10.3.12.4 Broken Master Detect

The TX4938 On-chip PCI Bus Arbiter has a function for automatically detecting broken masters.

If the PCI Bus Master requests and is granted the bus when the PCI Bus is in the Idle state, this master must assert the FRAME* signal within 16 PCI block cycles and start a transaction. The PCI Bus Arbiter recognizes any device that breaks this rule as a broken bus master and removes that device from the bus arbitration sequence.

This detection function is enabled when the Broken Master Check Enable bit (BMCEN) of the PCI Bus Arbiter Configuration Register (PBACFG) is set. When a broken master is detected, the Broken Master Detection bit (PBSTATUS.BMD) of the PCI Bus Arbiter Status Register is set and the bit in the PCI Bus Arbiter Broken Master Register (PBABM) that corresponds to that master is set. Then it also becomes possible to report an interrupt.

10.3.12.5 Special Programming

There may be some devices among PCI bus masters that operate differently from typical PCI devices. PCI devices with the following characteristics can be made usable by changing the programming of the PCI bus arbiter.

1. Bus masters that can not re-assert REQ unless GNT is once deasserted after deasserting REQ
   - Assign the bus master to a request port other than Port A through the PBAREQPORT register (at 0xD100). (Assign the TX4938 to Port A.)
   - Enable the Fixed Parked Master (FIXPA) bit in the PBACFG register (at 0xD104).

2. Bus masters that initiate a PCI transaction even when the deassertion of GNT has taken away their bus mastership before the start of the transaction
   - Assign the bus master to request port A, B, C or D through the PBAREQPORT register (at 0xD100).

For example, a bus master with both of the above characteristics can be used by configuring the PCI bus arbiter as follows:

Set the internal PCI bus arbiter to the fixed parked master.

Assign the TX4938 to request port A.

Assign the bus master to request port B.

If this bus master is connected to REQ[3] and broken master checking is to be enabled, values to be written to the PBACFG and PBAREQPORT registers are as follows:

PBACFG (at 0xD104): 0x0000000B

PBAREQPORT (at 0xD100): 0x73546210

10.3.13 PCI Boot

Setting the configuration during boot up (ADDR[8:6]) makes it possible to set the reset exception vector address of the TX49/H3 core to PCI Bus address 0x00_BFC0_0000.

Two windows of the memory space from the G-Bus to the PCI Bus space are used when in the PCI Boot mode. The defaults of several registers are changed as indicated below.

- G-Bus base address (G2GBASE): 0x0_1FC0_0000
- Space size (G2PM2MASK): 4 MB
- PCI Bus base address (G2PM2PBASE): 0x00_BFC0_0000
- Initiator Memory Space 2 Enable (PCICCFG.G2PM2EN): 1
- Bus Master bit (PCISTATUS.BM) [Only when in the Host mode] 1
- Target Configuration Access Ready (PCICSTAUTS.TCAR) [Only when in the Satellite mode] 1

Also, the on-chip PCI Bus Arbiter cannot be used when the PCI Boot mode is being used while in the Satellite mode.

10.3.14 Set Configuration Space

In Table 10.5.1, the values for the registers inside the PCI Configuration Space Register that have a gray background can be rewritten using one of the two following methods.

10.3.14.1 Set the Configuration Space Using EEPROM

Load values during Reset by connecting standard 93C46/93C48 EEPROM to a dedicated port.

The PCI Controller reads 16-bit half-word data for address 2n (n: 0, 1, 2, …, 31) of the PCI Configuration Space from EEPROM address (2n + 2 - 4(n mod 2)). Also, 16-bit data is read in order from the upper bits to the lower bits. The EEPROM values that correspond to the registers in Table 10.5.1 that have a white background are “don’t care”.

10.3.14.2 Set the Configuration Space Using Software Reset

By using the following procedure, it is possible to use the software to set the configuration space without using EEPROM.

(1) Set the value to be loaded in the Configuration Data 0 Register (PCICDATA0), the Configuration Data 1 Register (PCICDATA1), the Configuration Data 2 Register (PCICDATA2), and the Configuration Data 3 Register (PCICDATA3).

(2) Set the Load Configuration Data Register bit (LCFG) of the PCI Controller Configuration Register (PCICCFG) and the Software Reset bit (SRST).

(3) Clear the Software Reset bit (PCICCFG.SRST) at least four PCI Bus clock cycles later. This starts loading the data.

After these processes are complete, please set the Target Configuration Access Ready bit (PCICCFG.TCAR) of the PCI Controller Configuration Register to be able to accept access to the PCI Configuration space.

10.3.15 PCI Clock

The PCI bus signals are synchronized by the PCI clock applied to the PCICLKIN pin. Therefore, in PCI clock output mode, the PCI output clock must be connected to the PCICLKIN pin.

10.4 PCI Controller Control Register

Table 10.4.1 lists the registers contained in the PCI Controller Control Register. Parentheses in the register names indicate the corresponding PCI Configuration Space Register.

Table 10.4.1 PCI Controller Control Register (1/2)

Section	Address	Size	Mnemonic	Register Name
10.4.1	0xD000	32	PCIID	ID Register (Device ID, Vendor ID)
10.4.2	0xD004	32	PCISTATUS	PCI Status, Command Register (Status, Command)
10.4.3	0xD008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
10.4.4	0xD00C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
10.4.5	0xD010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
10.4.6	0xD014	32	P2GM0PUBASE	P2G Memory Space 0 PCI Upper Base Address Register (Base Address 0 Upper)
10.4.7	0xD018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
10.4.8	0xD01C	32	P2GM1PUBASE	P2G Memory Space 1 PCI Upper Base Address Register (Base Address 1 Upper)
10.4.9	0xD020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
10.4.10	0xD024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
10.4.11	0xD02C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
10.4.12	0xD034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
10.4.13	0xD03C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
10.4.14	0xD040	32	G2PTOCNT	G2P Timeout Count Register (Retry Timeout Value, TRDY Timeout Value)
10.4.15	0xD080	32	G2PSTATUS	G2P Status Register
10.4.16	0xD084	32	G2PMASK	G2P Interrupt Mask Register
10.4.17	0xD088	32	PCISSTATUS	Satellite Mode PCI Status Register (Status, PMCSR)
10.4.18	0xD08C	32	PCIMASK	PCI Status Interrupt Mask Register
10.4.19	0xD090	32	P2GCFG	P2G Configuration Register
10.4.20	0xD094	32	P2GSTATUS	P2G Status Register
10.4.21	0xD098	32	P2GMASK	P2G Interrupt Mask Register
10.4.22	0xD09C	32	P2GCCMD	P2G Current Command Register
10.4.23	0xD100	32	PBAREQPORT	PCI Bus Arbiter Request Port Register
10.4.24	0xD104	32	PBACFG	PCI Bus Arbiter Configuration Register
10.4.25	0xD108	32	PBASTATUS	PCI Bus Arbiter Status Register
10.4.26	0xD10C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
10.4.27	0xD110	32	PBABM	PCI Bus Arbiter Broken Master Register
10.4.28	0xD114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for diagnositics)
10.4.29	0xD118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for diagnostics)
10.4.30	0xD11C	32	PBACSTATE	PCI Bus Arbiter Currrent State Register (for diagnostics)
10.4.31	0xD120	64	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
10.4.32	0xD128	64	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
10.4.33	0xD130	64	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
10.4.34	0xD138	64	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
10.4.35	0xD140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
10.4.36	0xD144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
10.4.37	0xD148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
10.4.38	0xD14C	32	G2PIOMASK	G2P I/O Space Address Mask Register
10.4.39	0xD150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
10.4.40	0xD158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register

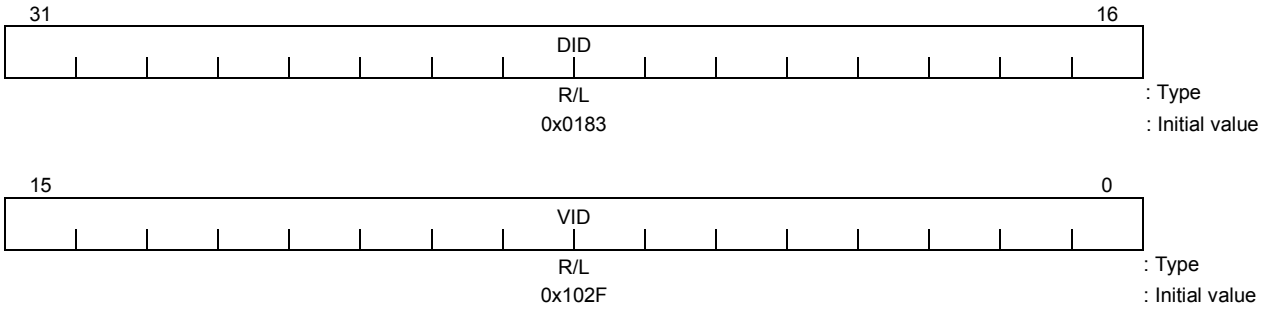
Table 10.4.1 PCI Controller Control Register (2/2)

Section	Address	Size	Mnemonic	Register Name
10.4.41	0xD160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
10.4.42	0xD168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
10.4.43	0xD170	32	PCICCFG	PCI Controller Configuration Register
10.4.44	0xD174	32	PCICSTATUS	PCI Controller Status Register
10.4.45	0xD178	32	PCICMASK	PCI Controller Interrupt Mask Register
10.4.46	0xD180	64	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
10.4.47	0xD188	64	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
10.4.48	0xD190	64	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
10.4.49	0xD198	64	P2GIOGBASE	P2G I/O Space G-Bus Base Address Register
10.4.50	0xD1A0	32	G2PCFGADRS	G2P Configuration Address Register
10.4.51	0xD1A4	32	G2PCFGDATA	G2P Configuration Data Register
10.4.52	0xD1C8	32	G2PINTACK	G2P Interrupt Acknowldge Data Register
10.4.53	0xD1CC	32	G2PSPC	G2P Special Cycle Data Register
10.4.54	0xD1D0	32	PCICDATA0	Configuration Data 0 Register
10.4.55	0xD1D4	32	PCICDATA1	Configuration Data 1 Register
10.4.56	0xD1D8	32	PCICDATA2	Configuration Data 2 Register
10.4.57	0xD1DC	32	PCICDATA3	Configuration Data 3 Register
10.4.58	0xD200	64	PDMCA	PDMAC Chain Address Register
10.4.59	0xD208	64	PDMGA	PDMAC G-Bus Address Register
10.4.60	0xD210	64	PDMPA	PDMAC PCI Bus Address Register
10.4.61	0xD218	64	PDMCTR	PDMAC Count Register
10.4.62	0xD220	64	PDMCFG	PDMAC Configuration Register
10.4.63	0xD228	64	PDMSTATUS	PDMAC Status Register

10.4.1 ID Register (PCIID) 0xD000[HH5]

The Device ID field corresponds to the Device ID Register in the PCI Configuration Space, and the Vendor ID field corresponds to the Vendor ID register of the PCI Configuration Space.

This register cannot be access when in the Satellite mode.



Bits	Mnemonic	Field Name	Description	Read/Write
31:16	DID	Device ID	Device ID (Default: 0x0183) This register indicates the ID that is allocated to a device. The ID can be changed by loading data from a configuration EEPROM during initialization.	R/L
15:0	VID	Vendor ID	Vendor ID (Default: 0x102F) This register indicates the device product that is allocated by PCI SIG. The product allocation can be changed by loading data from a configuration EEPROM during initialization.	R/L

Figure 10.4.1 ID Registers

10.4.2 PCI Status, Command Register (PCISTATUS) 0xD004

The upper 16 bits correspond to the Status Register in the PCI Configuration Space, and the lower 16 bits correspond to the Command Register in the PCI Configuration Space.

This register cannot be accessed when in the Satellite mode. However, it is possible to read some values of the upper 16 bits from the Satellite Mode PCI Status Register (PCISSTATUS).

31	30	29	28	27	26	25	24	23	22	21	20	19	16	
DPE	SSE	RMA	RTA	STA	DT		MDPE	FBBCP	Reserved	66MCP	CL	Reserved		
R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R		R/W1C	R		R	R			: Type
0	0	0	0	0	01		0	1		1	1			: Initial value

15	10	9	8	7	6	5	4	3	2	1	0	
Reserved		FBBEN	SEREN	STPC	PEREN	VPS	MWIEN	SC	BM	MEMSP	IOSP	
		R/W	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	: Type
		0	0	0	0	0	0	0	0/1	0	0	: Initial value

Bit	Mnemonic	Field Name	Description	Read/Write
31	DPE	Detected Parity Error	Detected Parity Error (Default: 0) Indicates that a parity error was detected. A parity error is detected in the three following situtations: • Detected a data parity error as the Read command PCI initiator. • Detected a data parity error as the Write command PCI target. • Detected an address parity error. This bit is set regardless of the setting of the Parity Error Response bit (PCISTATUS.PEREN) of the PCI Status, Command Register. 1: Detected a parity error. 0: Did not detect a parity error.	R/W1C
30	SSE	Signaled System Error	Signaled System Error (Default: 0) Detects either an address parity error or a special cycle data parity error. This bit is set when the SERR* signal is asserted. 1: Asserted the SERR* signal 0: Did not assert the SERR* signal.	R/W1C
29	RMA	Received Master Abort	Received Master Abort (Default: 0) This bit is set when a Master Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI initiator (except for special cycles). 1: Transaction was aborted by a Master Abort. 0: Transaction was not aborted by a Master Abort.	R/W1C
28	RTA	Received Target Abort	Received Target Abort (Default: 0) This bit is set when a Target Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI initiator. 1: Transaction was aborted by a Target Abort. 0: Transaction was not aborted by a Target Abort.	R/W1C
27	STA	Signaled Target Abort	Signaled Target Abort (Default: 0) This bit is set when a Target Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI target. 1: Bus transaction was aborted by a Target Abort. 0: Bus transaction was not aborted by a Target Abort.	R/W1C
26:25	DT	DEVSEL Timing	DEVSEL Timing (Fixed Value: 01) Three DEVSEL assert timings are defined in the PCI 2.2 Specifications: 00b = Fast; 01b = Medium; 10b = Slow; 11b = Reserved). With the exception of Read Configuration and Write Configuration, when the PCI Controller is the PCI target, the DEVSEL signal is asserted to a certain bus command and indicates the slowest speed for responding to the PCI Bus Master.	R

Figure 10.4.2 PCI Status, Command Register (1/3)

Bit	Mnemonic	Field Name	Description	Read/Write
24	MDPE	Master Data Parity Error	Master Data Parity Error (Default: 0) Indicates the a parity error occurred when the PCI Controller is the PCI initiator. This bit is not set when the PCI Controller is the target. This bit is set when all of the three following conditions are met. • It has been detected that the PERR* signal was set either directly or indirectly. • The PCI Controller is the Bus Master for a PCI Bus transaction during which an error occurred. • The Parity Error Response bit of the PCI Status Command Register (PCISTATUS.PEREN) has been set.	R/W1C
23	FBBCP	Fast Back-to-Back Capable	Fast Back-to-Back Capable (Fixed Value: 1) Indicates whether target access of a fast back-to-back transaction can be accepted. Is fixed to “1”.	R
22		Reserved		—
21	66MCP	66 MHz Capable	66 MHz Capable (Fixed Value: 1) Indicates the 66 MHz operation is possible. Is fixed to “1”.	R
20	CL	Capabilities List	Capabilities List (Fixed Value: 1) Indicates that the capabilities list is being implemented. Is fixed to “1”.	R
19:10		Reserved		—
9	FBBEN	Fast Back-to-Back Enable	Fast Back-to-Back Enable (Default: 0) Indicates that issuing of fast back-to-back transactions has been enabled. 1: Enable 0: Disable	R/W
8	SEREN	SERR* Enable	SERR* Enable (Default: 0) Enables/Disables the SERR* signal. The SERR* signal reports that either a PCI Bus address parity error or a special cycle data parity error was detected. The SERR* signal is only asserted when the Parity Error Response bit is set and this bit is set. 1: Enable 0: Disable	R/W
7	STPC	Stepping Control	Stepping Control (Fixed Value: 0) Indicates that stepping control is not being supported.	R
6	PEREN	Parity Error Response	Parity Error Response (Default 0) Sets operation when a PCI address/data parity error is detected. A parity error response (either when the Parity Error Response bit (PCISTATUS.PEREN) of the PERR* Signal Assert or PCI Status, Command Register is set, or the SERR* signal is asserted) is performed only when this bit is set. When this bit is cleared, the PCI Controller ignores all parity errors and continues the transaction process as if the parity of that transaction was correct. 1: Parity error response is performed. 0: Parity error response is not performed.	R/W
5	VPS	VGA Palette Snoop	VGA Palette Snoop (Fixed Value: 0) Indicates that the VGA palette snoop function is not supported.	R
4	MWIEN	Memory Write and Invalidate Enable	Memory Write and Invalidate Enable (Default: 0) Controls whether to use the Memory Write and Invalidate command instead of the Memory Write command when the PCI Controller is the initiator.	R/W
3	SC	Special Cycles	Special Cycles (Fixed Value: 0) Indicates that special cycles will not be accepted as PCI targets.	R
2	BM	Bus Master	Bus Master (Default: 0/1) The default is only “1” when in the PCI Boot mode and in the Host mode. 1: Operates as the Bus Master. 0: Does not operate as the Bus Master.	R/W

Figure 10.4.2 PCI Status, Command Register (2/3)

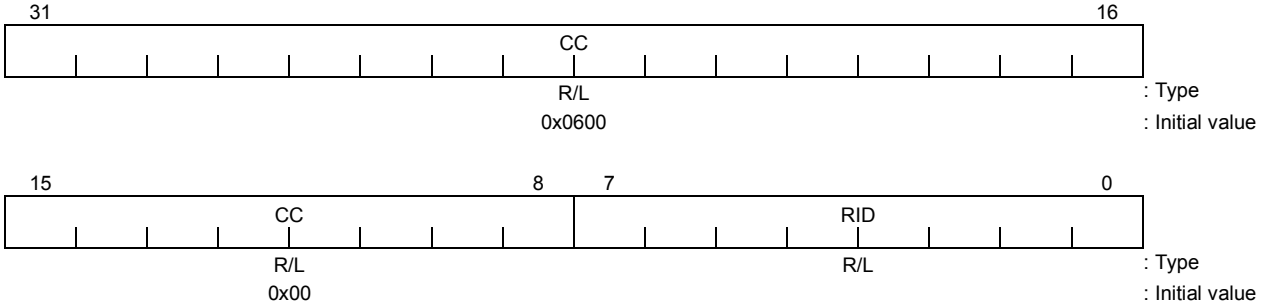
Bit	Mnemonic	Field Name	Description	Read/Write
1	MEMSP	Memory Space	Memory Space (Default: 0) 1: Respond to PCI memory access. 0: Do not respond to PCI memory access.	R/W
0	IOSP	I/O Space	I/O Space (Default: 0) 1: Respond to PCI I/O access. 0: Do not respond to PCI I/O access.	R/W

Figure 10.4.2 PCI Status, Command Register (3/3)

10.4.3 Class Code, Revision ID Register (PCICCREV) 0xD008

The Class Code field corresponds to the Class Code Register of the PCI Configuration Space, and the Revision ID field corresponds to the Revision ID Register of the PCI Configuration Space.

This register cannot be accessed when in the Satellite mode.



Bits	Mnemonic	Field Name	Description	Read/Write
31:8	CC	Class Code	Class Code (Default: 0x060000) Classifies the device types. The default is 060000h, which defines the PCI Controller as a Host bridge device. It is possible to change the device type by loading data from Configuration EEPROM during initialization.	R/L
7:0	RID	Revision ID	Revision ID Indicates the device revision ID. Please contact our Engineering Department for the exact value. It is possible to change the revision ID by loading data from Configuration EEPROM during initialization.	R/L

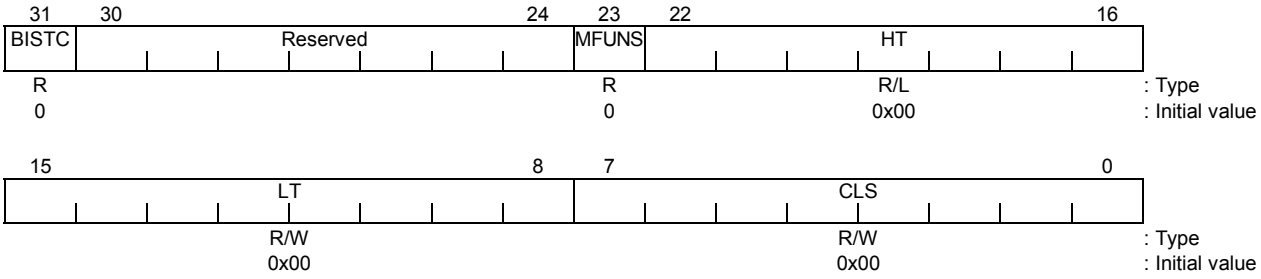
Figure 10.4.3 Class Code, Revision ID Register

10.4.4 PCI Configuration 1 Register (PCICFG1) 0xD00C

The following fields correspond to the following registers.

BIST field → BIST Register of the PCI Configuration Space
Header Type field → Header Type Register in the PCI Configuration Space
Latency Timer field → Latency Timer Register of the PCI Configuration Space
Cache Line Size field → Cache Line Size Register of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



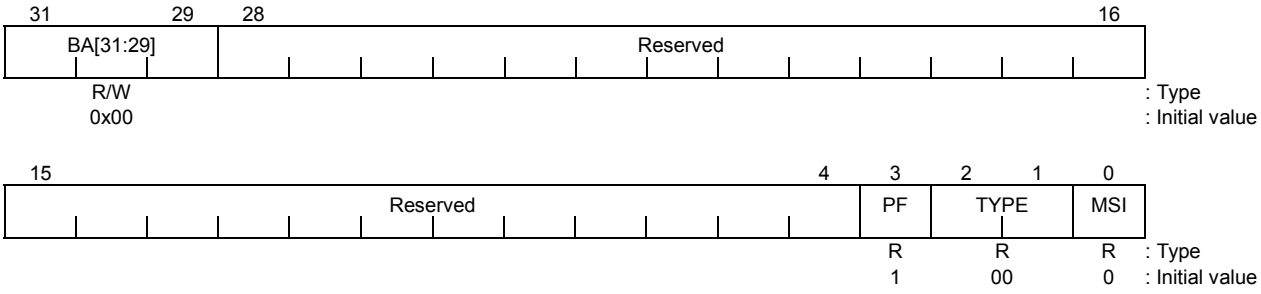
Bit	Mnemonic	Field Name	Description	Read/Write
31	BISTC	BIST Capable	BIST Capable (Fixed Value: 0) Indicates that the BIST function is not being supported.	R
30:24		Reserved		—
23	MFUNS	Multi-Function	Multi-Function (Fixed Value: 0) 0: Indicates that the device is a single-function device.	R
22:16	HT	Header Type	Header Type (Default: 0x00) Indicates the Header type. 0000000: Header Type 0 It is possible to change the header type by loading data from Configuration EEPROM during initialization.	R/L
15:8	LT	Latency Timer	Latency Timer (Default: 0x00) Sets the latency timer value. Specifies the PCI Bus clock count during which to abort access when the GNT* signal is deasserted during PCI access. Since the lower two bits are fixed to “0”, cycle counts can only be specified in multiples of 4.	R/W
7:0	CLS	Cache Line Size	Cache Line Size (Default: 0x00) Is used to select the PCI Bus command during a Burst Read transaction. See “10.3.3 Supported PCI Bus Commands)” for more information.	R/W

Figure 10.4.4 PCI Configuration 1 Register

10.4.5 P2G Memory Space 0 PCI Lower Base Address Register (P2GM0PLBASE) 0xD010

This register corresponds to the Memory Space 0 Lower Base Address Register at offset address 0x10 of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



Bit	Mnemonic	Field Name	Description	Read/Write
31:29	BA[31:29]	Base Address	Base Address (Default: 0x00) Sets the lower address of the PCI base address in Target Access Memory Space 0. The size of Memory Space 0 is fixed at 512 MB.	R/W
28:4		Reserved		—
3	PF	Prefetchable	Prefetchable (Fixed Value: 1) 1: Indicates that memory is prefetchable.	R
2:1	TYPE	Type	Type (Default: 00) 00: Indicades that an address is within a 32-bit address region.	R
0	MSI	Memory Space	Memory Space Indicator (Fixed Value: 0) 0: Indicates that this Base Address Register is for use by the PCI Memory Space.	R

Figure 10.4.5 P2G Memory Space 0 PCI Lower Base Address Register

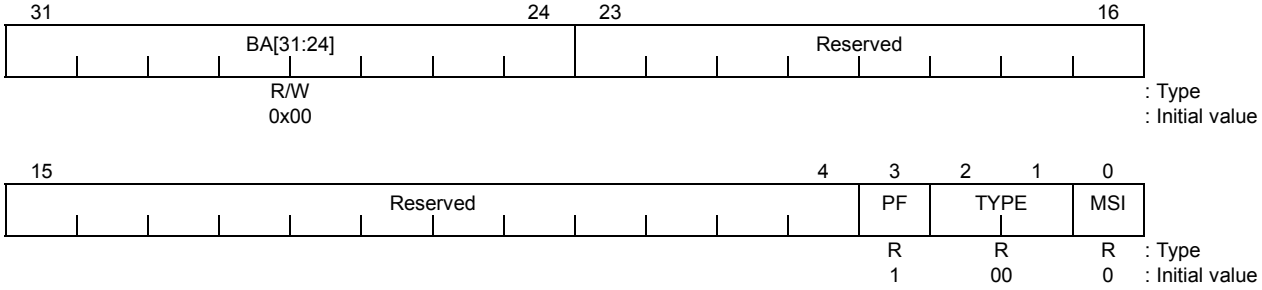
10.4.6 P2G Memory Space 0 PCI Upper Base Address Register (P2GM0PUBASE) 0xD014

This register is unused since the PCI Controller does not support the target dual-address cycle. It is forbidden to write to this register.

10.4.7 P2G Memory Space 1 PCI Lower Base Address Register (P2GM1PLBASE) 0xD018

This register corresponds to the Memory Space 1 Lower Base Address Register at offset address 0x18 of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



Bit	Mnemonic	Field Name	Description	Read/Write
31:24	BA[31:24]	Base Address	Base Address (Default: 0x00) Sets the lower address of the PCI base address in Target Access Memory Space 1. The size of Memory Space 1 is fixed at 16 MB.	R/W
23:4		Reserved		—
3	PF	Prefetchable	Prefetchable (Fixed Value: 1) 1: Indicates that memory is prefetchable.	R
2:1	TYPE	Type	Memory Type (Default: 00) 00: Indicates that memory is placed in the 32-bit address space.	R
0	MSI	Memory Space	Memory Space Indicator (Fixed Value: 0) 0: Indicates that this Base Address Register is for use by the PCI Memory Space.	R

Figure 10.4.6 P2G Memory Space 1 PCI Lower Base Address Register

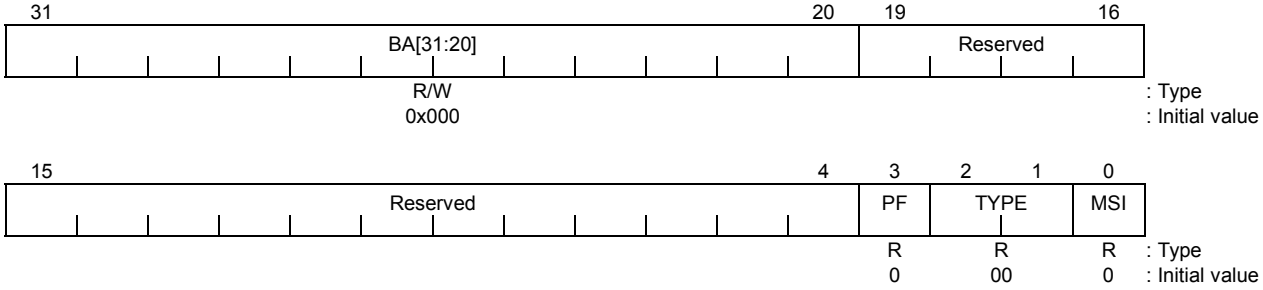
10.4.8 P2G Memory Space 1 PCI Upper Base Address Register (P2GM1PUBASE) 0xD01C

This register is unused since the PCI Controller does not support the target dual-address cycle. It is forbidden to write to this register.

10.4.9 P2G Memory Space 2 PCI Base Address Register (P2GM2PBASE) 0xD020

This register corresponds to the Memory Space 2 Base Address Register at offset address 0x20 of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



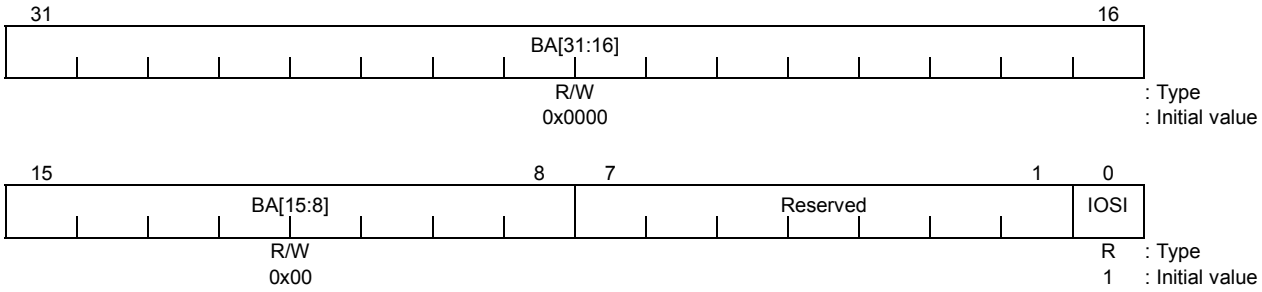
Bit	Mnemonic	Field Name	Description	Read/Write
31:20	BA[31:20]	Base Address	Base Address (Default: 0x00) Sets the PCI base address in Target Access Memory Space 2. The size of Memory Space 12 is fixed at 1 MB.	R/W
19:4		Reserved		—
3	PF	Prefetchable	Prefetchable (Fixed Value: 0) 0: Indicates that memory is not prefetchable.	R
2:1	TYPE	Type	Memory Type (Default: 00) 00: Indicates that an address is within a 32-bit address region.	R
0	MSI	Memory Space	Memory Space Indicator (Fixed Value: 0) 0: Indicates that this Base Address Register is for use by the PCI Memory Space.	R

Figure 10.4.7 P2G Memory Space 2 PCI Base Address Register

10.4.10 P2G I/O Space PCI Base Address Register (P2GIOPBASE) 0xD024

This register corresponds to the I/O Space Base Address at offset address 0x24 of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



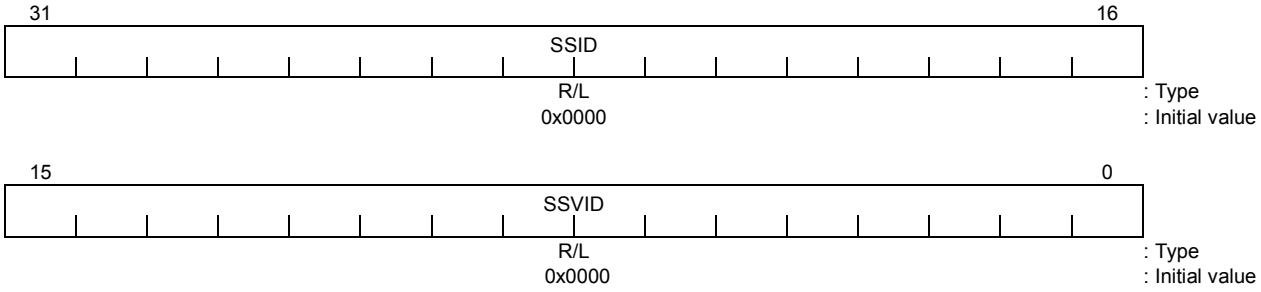
Bit	Mnemonic	Field Name	Description	Read/Write
31:8	BA[31:8]	Base Address	Base Address (Default: 0x00) Sets the PCI base address of the Target Access I/O Space. The size of this I/O space is fixed at 256 Bytes.	R/W
7:1		Reserved		—
0	IOSI	I/O Space	I/O Space Indicator (Fixed Value: 1) 1: Indicates that this Base Address Register is for use by the PCI I/O Space.	R

Figure 10.4.8 P2G I/O Space PCI Base Address Register

10.4.11 Subsystem ID Register (PCISID) 0xD02C

The Subsystem ID field corresponds to the Subsystem ID Register of the PCI Configuration Space, and the Subsystem Vendor ID field corresponds to the Subsystem Vendor ID Register of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



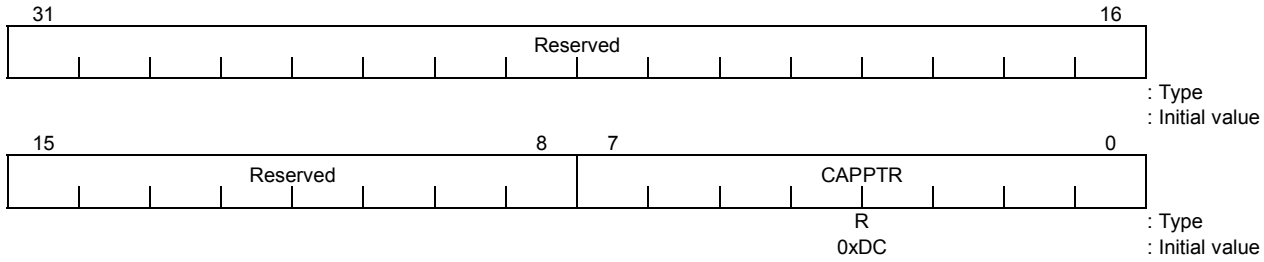
Bits	Mnemonic	Field Name	Description	Read/Write
31:16	SSID	Subsystem ID	Subsystem ID (Default: 0x0000) This register is used to acknowledge either a subsystem that has a PCI device or an add-in board. It is possible to change the Subsystem ID by loading data from Configuration EEPROM during initialization.	R/L
15:0	SSVID	Subsystem Vendor ID	Subsystem Vendor ID (Default: 0x0000) This register is used to acknowledge either a subsystem that has a PCI device or an add-in board. It is possible to change the Subsystem ID by loading data from Configuration EEPROM during initialization.	R/L

Figure 10.4.9 Subsystem ID Register

10.4.12 Capabilities Pointer Register (PCICAPPTR) 0xD034

The Capabilities Pointer field corresponds to the Capabilities Pointer Register of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



Bits	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7:0	CAPPTR	Capabilities Pointer	Capabilities Pointer (Fixed Value: 0xDC) Indicates as an offset value the starting address of the capabilities list that indicates extended functions.	R

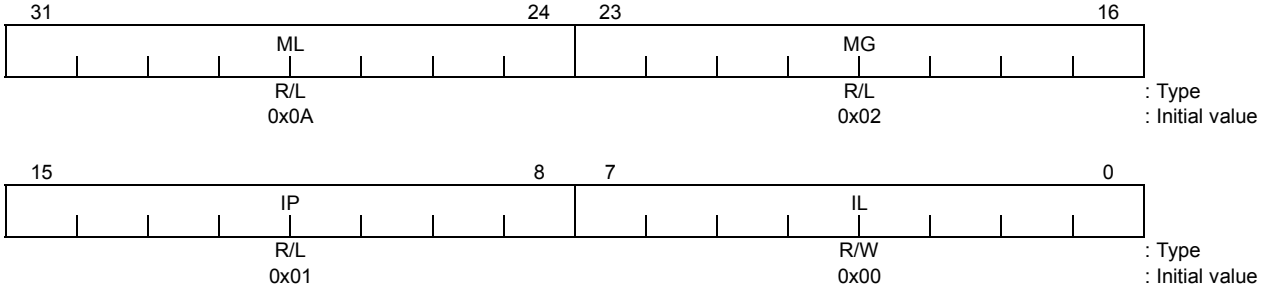
Figure 10.4.10 Capabilities Pointer Register

10.4.13 PCI Configuration 2 Register (PCICFG2) 0xD03C

The following fields correspond to the following registers:

Max. Latency field → Max_Lat Register of the PCI Configuration Space
Min. Grant field → Min_Gnt Register of the PCI Configuration Space
Interrupt Pin field → Interrupt Pin Register of the PCI Configuration Space
Interrupt Line field → Interrupt Line Register of the PCI Configuration Space

This register cannot be accessed when the PCI Controller is in the Satellite mode.



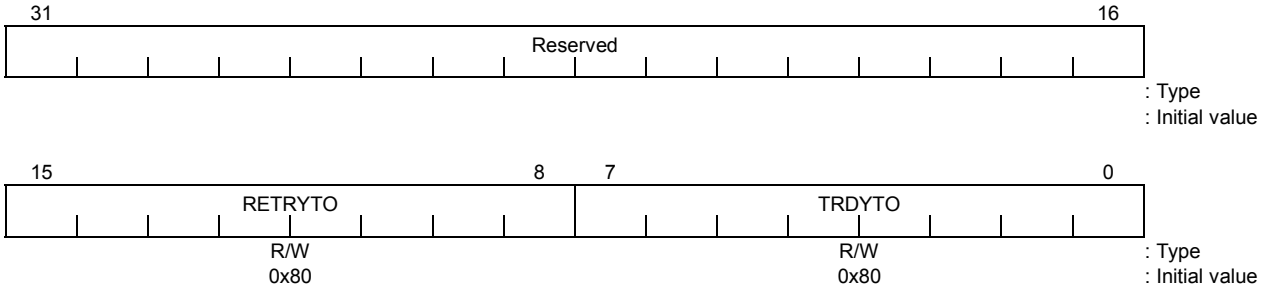
Bits	Mnemonic	Field Name	Description	Read/Write
31:24	ML	Maximum Latency	Max_Lat (Maximum Latency) (Default: 0x0A) 00h: Does not use this register to determine PCI Bus priority. 01h-FFh: Specifies the time interval for requesting bus ownership. In units of 250 ns, assuming the PCICLK is 33 MHz. It is possible to change the maximum latency by loading data from Configuration EEPROM during initialization.	R/L
23:16	MG	Minimum Grant	Min_Gnt (Minimum Grant) (Default: 0x02) 00h: Is not used to calculate the latency timer value. 01h-FFh: Sets the time required for Burst transfer. In units of 250 ns, assuming the PCICLK is 33 MHz. It is possible to change this value by loading data from Configuration EEPROM during initialization.	R/L
15:8	IP	Interrupt Pin	Interrupt Pin (Default: 0x01) Valid values: 00 - 04h 00h: Do not use interrupt signals. 01h: Use Interrupt signal INTA* 02h: Use Interrupt signal INTB* 03h: Use Interrupt signal INTC* 04h: Use Interrupt signal INTD* 05h - FFh: Reserved It is possible to change this value by loading data from Configuration EEPROM during initialization. When using either the REQ[2]* signal or the PIO signal to report an interrupt to an external device as the PCI device, please use EEPROM to set the connection with that device.	R/L
7:0	IL	Interrupt Line	Interrupt Line (Default: 0x00) This is a readable/writable 8-bit register. The software uses this register to indicate information such as the interrupt signal connection information. Operation of the TX4938 is not affected.	R/W

Figure 10.4.11 PCI Configuration 2 Register

10.4.14 G2P Timeout Count Register (G2PTOCNT) 0xD040

The Retry Timeout field corresponds to the Retry Timeout Value Register of the PCI Configuration Space, and the TRDY Timeout field corresponds to the TRDY Timeout Value Register of the PCI Configuration Space.

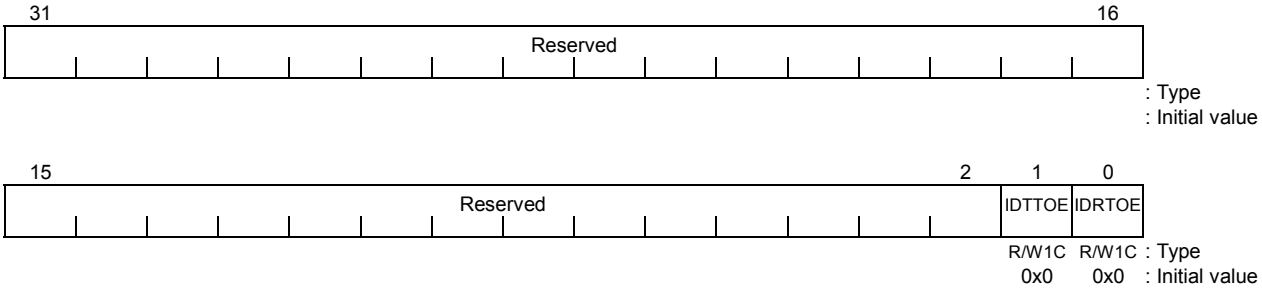
This register cannot be accessed when the PCI Controller is in the Satellite mode.



Bits	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15:8	RETRYTO	Retry Timeout	Retry Time Out (Default: 0x80) Sets the maximum number of retries to accept when operating as the initiator on the PCI Bus. Ends with an error when receiving more retry terminations than the set maximum number. Setting a “0” disables this timeout function. Note: Generally, disable retry time-out detection by setting this field to zero. Some PCI devices invoke more than 128 retries at normal times.	R/W
7:0	TRDYTO	TRDY Timeout	TRDY Time Out (Default: 0x80) Sets the maximum value of the time to wait for assertion of the TRDY* signal when operating as the initiator on the PCI Bus. Setting a “0” disables this timeout function. Note: Generally, disable TRDY time-out detection by setting this field to zero. Some PCI devices exhibit a TRDY delay longer than 128 PCI clocks at normal times.	R/W

Figure 10.4.12 G2P Timeout Count Register

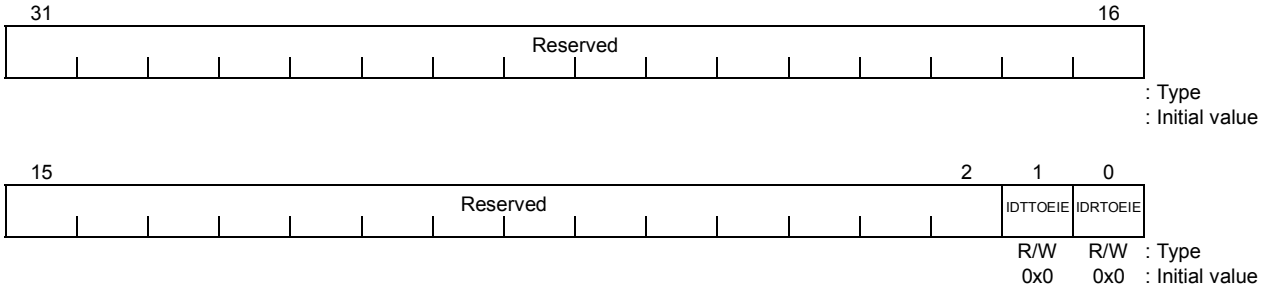
10.4.15 G2P Status Register (G2PSTATUS) 0xD080



Bit	Mnemonic	Field Name	Description	Read/Write
31:2		Reserved		—
1	IDTTOE	TRDY Timeout Error	Initiator Detected TRDY Time Out Error (Default: 0x0) This bit is set when the initiator detects a TRDY timeout.	R/W1C
0	IDRTOE	Retry Timeout Error	Initiator Detected Retry Time Out Error (Default: 0x0) This bit is set when the initiator detects a Retry timeout.	R/W1C

Figure 10.4.13 G2P Status Register

10.4.16 G2P Interrupt Mask Register (G2PMASK) 0xD084

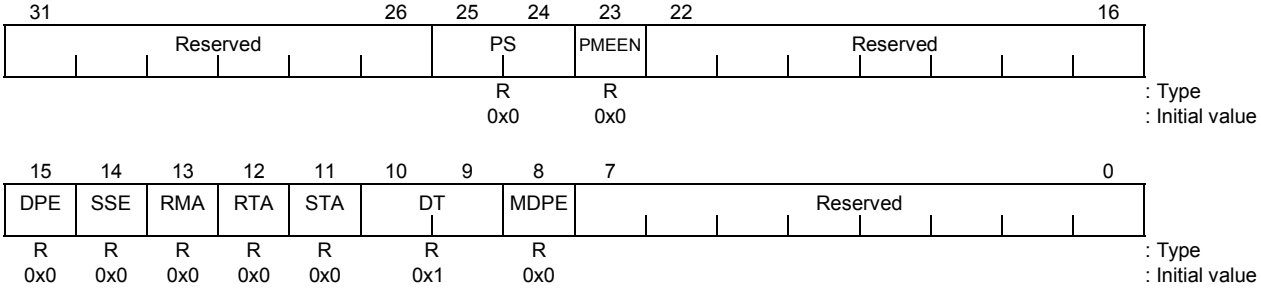


Bit	Mnemonic	Field Name	Description	Read/Write
31:2		Reserved		—
1	IDTTOEIE	TRDY Timeout Error Interrupt Enable	Initiator Detected TRDY Time Out Interrupt Enable (Default: 0x0) The initiator generates an interrupt when it detects a TRDY timeout. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
0	IDRTOEIE	Retry Timeout Error Interrupt Enable	Initiator Detected Retry Time Out Interrupt Enable (Default: 0x0) The initiator generates an interrupt when it detects a Retry timeout. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W

Figure 10.4.14 G2P Interrupt Mask Register

10.4.17 Satellite Mode PCI Status Register (PCISSTATUS) 0xD088

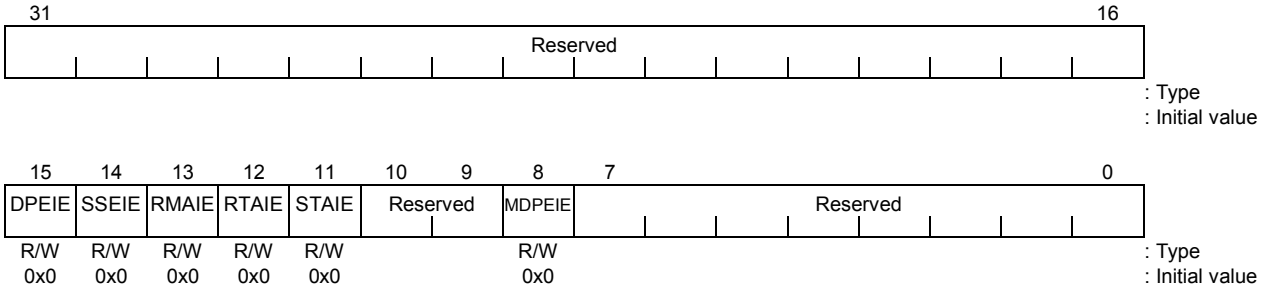
The PCI Status, Command Register (PCISTATUS) or the PMCSR Register of the Configuration Space cannot be accessed when the PCI Controller is in the Satellite mode. It is possible however to read values from either of these registers.



Bit	Mnemonic	Field Name	Description	Read/Write
31:24		Reserved		—
25:24	PS	Power State	PowerState (Default: 0x0) This is a shadow register of the PowerState field in the PMCSR Register. Note: Read this field in the following procedures. If other procedures are used, incorrect data may be read. (1) General procedures After checking the P2GSTATUS.PMSC bit is set, read the PS field. (2) Procedures to read at any time To read PS field directly, but not using the procedures shown above (1), read the PS field twice consecutively. Use the value if the same value is read.	R
23	PMEEN	PME Enable	PME_En (Default: 0x0) This is a shadow register of the PME_En bit of the PMCSR Register.	R
22:16		Reserved		—
15	DPE	Detected Parity Error	Detected Parity Error (Default: 0x0) This is a shadow register of the PCISTATUS.DPE bit.	R
14	SSE	Signaled System Error	Signaled System Error (Default: 0x0) This is a shadow register of the PCISTATUS.SSE bit.	R
13	RMA	Received Master Abort	Received Master Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RMA bit.	R
12	RTA	Received Target Abort	Received Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RTA bit.	R
11	STA	Signaled Target Abort	Signaled Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.STA bit.	R
10:9	DT	Set DEVSEL Timing	DEVSEL Timing (Fixed Value: 0x1) This is a shadow register of the PCISTATUS.DT field.	R
8	MDPE	Data Parity Detected	Master Data Parity Error Detected (Default: 0x0) This is a shadow register of the PCISTATUS.MDPE bit.	R
7:0		Reserved		—

Figure 10.4.15 Satellite Mode PCI Status Register

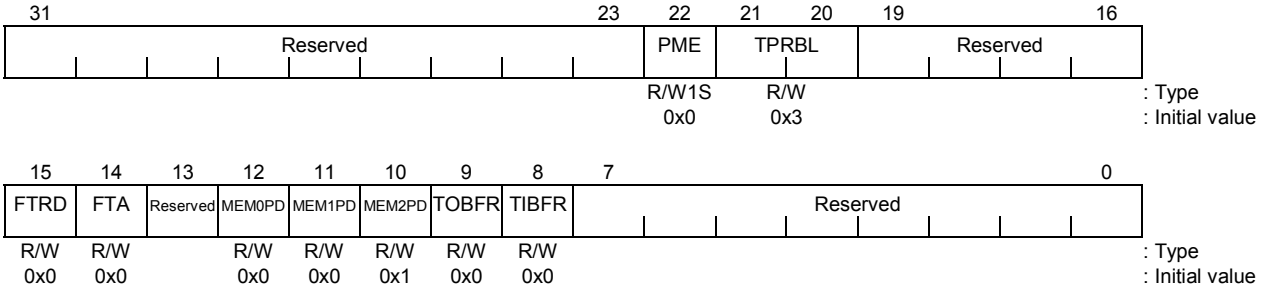
10.4.18 PCI Status Interrupt Mask Register (PCIMASK) 0xD08C



Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	DPEIE	Detected Parity Error Interrupt Enable	Detected Parity Error Interrupt Enable (Default: 0x0) Generates an interrupt when a parity error is detected. Usually, this interrupt is masked and a Master Data Parity error signals the error to the system. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
14	SSEIE	Signaled System Error Interrupt Enable	Signaled System Error Interrupt Enable (Default: 0x0) Generates an interrupt when a system error is signaled. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
13	RMAIE	Received Master Abort Interrupt Enable	Received Master Abort Interrupt Enable (Default: 0x0) Generates an interrupt when a Master Abort is received. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
12	RTAIE	Received Target Abort Interrupt Enable	Received Target Abort Interrupt Enable (Default: 0x0) Generates an interrupt when a Target Abort is received. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
11	STAIE	Signaled Target Abort Interrupt Enable	Signaled Target Abort Interrupt Enable (Default: 0x0) Generates an interrupt when a Target Abort is signaled. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
13	RMA	Received Master Abort	Received Master Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RMA bit.	R/W
12	RTA	Received Target Abort	Received Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RTA bit.	R/W
11	STA	Signaled Target Abort	Signaled Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.STA bit.	R/W
10:9		Reseved		—
8	MDPEIE	Master Data Parity Detected Interrupt Enable	Master Data Parity Detected Interrupt Enable (Default: 0x0) Generates an interrupt when data parity is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
7:0		Reserved		—

Figure 10.4.16 PCI Status Interrupt Mask Register

10.4.19 P2G Configuration Register (P2GCFG) 0xD090



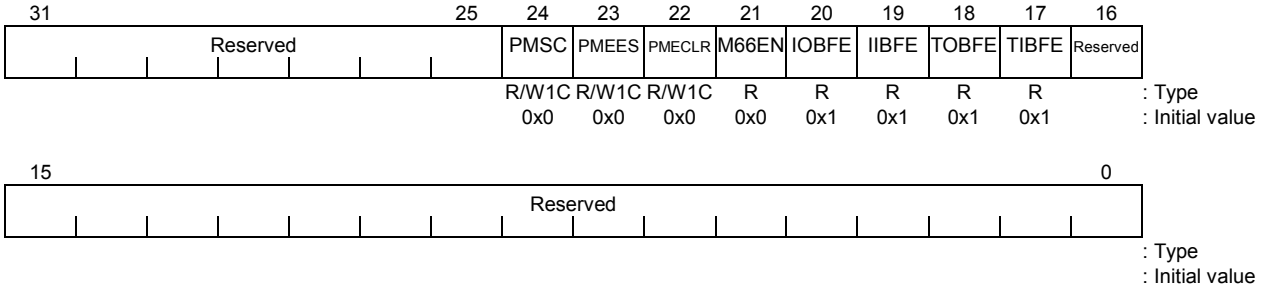
Bit	Mnemonic	Field Name	Description	Read/Write
31:23		Reserved		—
22	PME	PME	PME (Default: 0x0) When the PCI Controller is in the Satellite mode, writing “1” to this bit signals a PME (Power Management Event) to the PCI Host device. The PME* signal is asserted if the PME_Status bit of the PMCSR Register is set and the PME_En bit of the PMCSR Register is set. This bit is cleared when the PCI Host device writes a “1” to the PME_Status bit of the PMCSR Register. This bit is invalid when the PCI Contoller is in the Host mode since the PME* signal is an input signal.	R/W1S
21:20	TPRBL	Target Prefetch Read Burst Length	Target Prefetch Read Burst Length (Default: 0x3) These bits set the number of DWORDS (32-bit words) to be read into the data FIFO when prefetching is valid during a target memory Read operation. Extra data transferred to the data FIFO is deleted when performing a memory Read operation of a PCI Bus transfer that is smaller than the set size. This setting is invalid when prefetching is disabled. 0x00: Access and transfer each 2 DWORDs of data to the target read FIFO. 0x01: Access and transfer each 4 DWORDs of data to the target read FIFO. 0x10: Access and transfer each 6 DWORDs of data to the target read FIFO. 0x11: Access and transfer each 8 DWORDs of data to the target read FIFO.	R/W
19:16		Reserved		—
15	FTRD	Force Target Retry/Disconnect	Force Target Retry/Disconnect (Default: 0x0) The PCI Controller executes Retry Termination on a PCI Read access transaction if this bit is set to “1”. This is a diagnostic function.	R/W
14	FTA	Force Target Abort	Force Target Abort (Default: 0x0) The PCI Controller executes a Target Abort on a PCI Read access transaction if this bit is set to “1”. This is a diagnostic function.	R/W
13		Reserved		—
12	MEM0PD	Memory 0 Window Prefetch Disable	Memory 0 Window Prefetch Disable (Default: 0x0) Prefetching during a G-Bus Burst Read transfer cycle to the Memory 0 Space is disabled when this bit is set to “1”. PCI Burst Read transactions are not supported when prefetching is disabled. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change. We recommend using the default setting when the PCI Controller is in the Satellite mode.	R/W

Figure 10.4.17 P2G Configuration Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
11	MEM1PD	Memory 1 Window Prefetch Disable	Memory 1 Window Prefetch Disable (Default: 0x0) Prefetching during a G-Bus Burst Read transfer cycle to the Memory 1 Space is disabled when this bit is set to “1”. PCI Burst Read transactions are not supported when prefetching is disabled. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change. We recommend using the default setting when the PCI Controller is in the Satellite mode.	R/W
10	MEM2PD	Memory 2 Window Space Prefetch Disable	Memory 2 Window Prefetch Disable (Default: 0x1) Prefetching during a G-Bus Burst Read transfer cycle to the Memory 2 Space is disabled when this bit is set to “1”. PCI Burst Read transactions are not supported when prefetching is disabled. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change. We recommend using the default setting when the PCI Controller is in the Satellite mode.	R/W
9	TOBFR	Target Out-Bound FIFO Reset	Target Out-Bound FIFO Reset (Default: 0x0) The PCI Controller flushes the CORE internal Target Out-Bound FIFO when “1” is written to this bit. This bit always reads out “0” when it is read. This is a diagnostic function.	R/W
8	TIBFR	Target In-Bound FIFO Reset	Target In-Bound FIFO Reset (Default: 0x0) The PCI Controller flushes the CORE internal Target In-Bound FIFO when “1” is written to this bit. This bit always read out “0” when it is read. This is a diagnostic function.	R/W
7:0		Reserved		—

Figure 10.4.17 P2G Configuration Register (2/2)

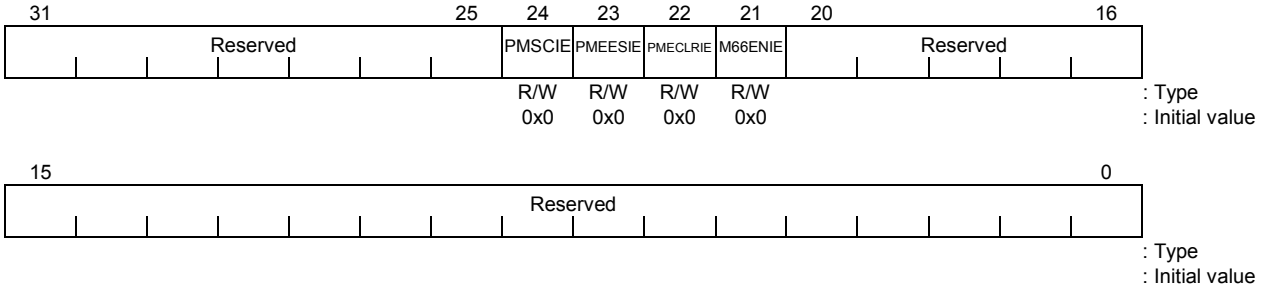
10.4.20 P2G Status Register (P2GSTATUS) 0xD094



Bit	Mnemonic	Field Name	Description	Read/Write
31:25		Reserved		—
24	PMSC	PM State Change Detected	Power Management State Change (Default: 0x0) "1" is set to this bit when the PowerState field of the Power Management Register (PMCSR) is rewritten. This bit is cleared to "0" when a "1" is written to it. This bit is only valid when the PCI Controller is in the Satellite mode.	R/W1C
23	PMEES	PME_En Set Detected	PME_En Set (Default: 0x0) This bit is set to "1" when the PME_En bit of the PMCSR Register is set to "1". When this bit is set, it indicates that the PCI Master (Host) device enabled PME* signal output. 1: Indicates that the PME_En bit is set. 0: Indicates that the PME_En bit is not set. This bit is cleared to "0" when a "1" is written to it. This bit is only valid when the PCI Controller is in the Satellite mode.	R/W1C
22	PMECLR	PME Status Clear Detected	PME_Status Clear (Default: 0x0) This bit indicates that the PME_Status bit of the PMCSR Register was cleared. 1: Indicates that the PME_Status bit was cleared. 0: Indicates that the PME_Status bit was not cleared. This bit is cleared to "0" when a "1" is written to it. This bit is only valid when the PCI Controller is in the Satellite mode.	R/W1C
21	M66EN	66 MHz Drive Status	M66EN Status (Default: 0x0) This bit indicates the current status of the M66EN signal. This bit can only be read. Writes to this bit are invalid. 1: The M66EN signal is asserted. 0: The M66EN signal is deasserted.	R
20	IOBFE	Initiator Out-Bound FIFO Empty	Initiator Out-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Initiator Out-Bound FIFO is empty. 0: Indicates that the Initiator Out-Bound FIFO is not empty. This is a diagnostic function.	R
19	IIBFE	Initiator In-Bound FIFO Empty	Initiator In-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Initiator In-Bound FIFO is empty. 0: Indicates that the Initiator In-Bound FIFO is not empty. This is a diagnostic function.	R
18	TOBFE	Target Out-Bound FIFO Empty	Target Out-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Target Out-Bound FIFO is empty. 0: Indicates that the Target Out-Bound FIFO is not empty. This is a diagnostic function.	R
17	TIBFE	Target In-Bound FIFO Empty	Target In-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Target In-Bound FIFO is empty. 0: Indicates that the Target In-Bound FIFO is not empty. This is a diagnostic function.	R
16:0		Reserved		—

Figure 10.4.18 P2G Status Reigster

10.4.21 P2G Interrupt Mask Register (P2GMASK) 0xD098



Bit	Mnemonic	Field Name	Description	Read/Write
31:25		Reserved		—
24	PMSCIE	Power Management State Change Interrupt Enable	Power Management State Change Interrupt Enable (Default: 0x0) Generates an interrupt when the PowerState field of the Power Management Register (PMCSR) is rewritten. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
23	PMEESIE	PME_En Set Interrupt Enable	PME_En Set Interrupt Enable (Default: 0x0) Generates an interrupt when the PME_En bit of the PMCSR Register is set. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
22	PMECLRIE	PME Status Clear Interrupt Enable	PME_Status Clear Interrupt Enable (Default: 0x0) Generates an interrupt when the PME_Status bit of the PMCSR Register is cleared. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
21	M66ENIE	66 MHz Drive Interrupt Enable	M66EN Detected Interrupt Enable (Default: 0x0) Generates an M66EN interrupt when the PCI Controller is in the Satellite mode. Note: This bit must be masked in order to clear an M66EN interrupt since the M66EN bit of the P2GSTATUS Register itself cannot be cleared. When the PCI Controller is in the Host mode, M66EN interrupts are invalid and will not be signaled even if this bit is set to “1”. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
20:0		Reseved		—

Figure 10.4.19 P2G Interrupt Mask Register

10.4.22 P2G Current Command Register (P2GCCMD) 0xD09C

Bits	Mnemonic	Field Name	Description	Read/Write
31:4		Reserved		—
3:0	TCCMD	Target Current Command Register	Target Current Command (Default: 0x0) Indicates the PCI command within the target access process that is currently in progress. This is a diagnostic function.	R

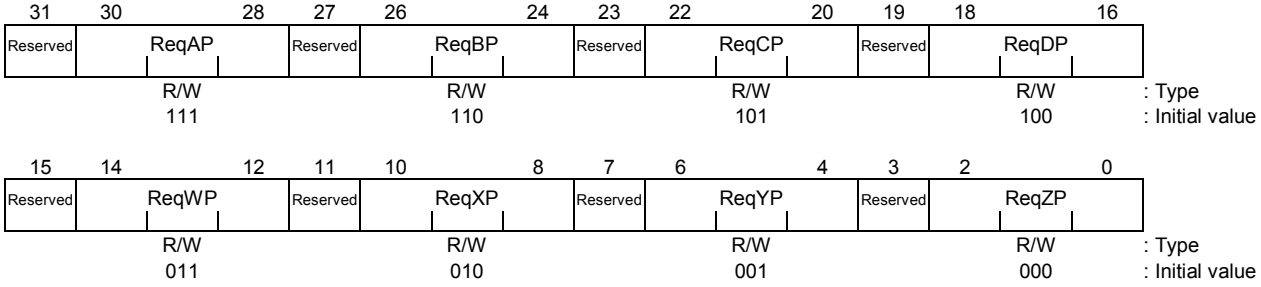
Figure 10.4.20 P2G Current Command Register

10.4.23 PCI Bus Arbiter Request Port Register (PBAREQPORT) 0xD100

This register sets the correlation between each PCI Bus request source (PCI Controller and REQ[3:0]) and each Internal PCI Bus Arbiter Request port (Master A - D, W - Z) (see Figure 10.3.8).

When changing the settings of this register, unused ports must be programmed to a reserved value. The eight non-reserved fields must be programmed to different values. After changing this register, the Broken Master Register (BM) value becomes invalid since the bit mapping changes.

This register is only valid when using the on-chip PCI Bus Arbiter.


31 30 28 27 26 24 23 22 20 19 18 16
Reserved ReqAP Reserved ReqBP Reserved ReqCP Reserved ReqDP
R/W R/W R/W R/W : Type
111 110 101 100 : Initial value
15 14 12 11 10 8 7 6 4 3 2 0
Reserved ReqWP Reserved ReqXP Reserved ReqYP Reserved ReqZP
R/W R/W R/W R/W : Type
011 010 001 000 : Initial value

Bit	Mnemonic	Field Name	Description	Read/Write
31		Reserved		—
30:28	ReqAP	Request A Port	Request A Port (Default: 111) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request A Port (Master A). 111: Makes the PCI Controller Master A. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master A. 010: Makes REQ*[2] Master A. 001: Makes REQ*[1] Master A. 000: Makes REQ*[0] Master A.	R/W
27		Reserved		—
26:24	ReqBP	Request B Port	Request B Port (Default: 110) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request B Port (Master B). 111: Makes the PCI Controller Master B. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master B. 010: Makes REQ*[2] Master B. 001: Makes REQ*[1] Master B. 000: Makes REQ*[0] Master B.	R/W
23		Reserved		—
22:20	ReqCP	Request C Port	Request C Port (Default: 101) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request C Port (Master C). 111: Makes the PCI Controller Master C. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master C. 010: Makes REQ*[2] Master C. 001: Makes REQ*[1] Master C. 000: Makes REQ*[0] Master C.	R/W

Figure 10.4.21 PCI Bus Arbiter Request Port Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
19		Reserved		—
18:16	ReqDP	Request D Port	Request D Port (Default: 100) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request D Port (Master D). 111: Makes the PCI Controller Master D. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master D. 010: Makes REQ*[2] Master D. 001: Makes REQ*[1] Master D. 000: Makes REQ*[0] Master D.	R/W
15		Reserved		—
14:12	ReqWP	Request W Port	Request W Port (Default: 011) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request W Port (Master W). 111: Makes the PCI Controller Master W. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master W. 010: Makes REQ*[2] Master W. 001: Makes REQ*[1] Master W. 000: Makes REQ*[0] Master W.	R/W
11		Reserved		—
10:8	ReqXP	Request X Port	Request X Port (Default: 010) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request X Port (Port X). 111: Makes the PCI Controller Master X. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master X. 010: Makes REQ*[2] Master X. 001: Makes REQ*[1] Master X. 000: Makes REQ*[0] Master X.	R/W
7		Reserved		—
6:4	ReqYP	Request Y Port	Request Y Port (Default: 001) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request Y Port (Port Y). 111: Makes the PCI Controller Master Y. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master Y. 010: Makes REQ*[2] Master Y. 001: Makes REQ*[1] Master Y. 000: Makes REQ*[0] Master Y.	R/W
3		Reserved		—
2:0	ReqZP	Request Z Port	Request Z Port (Default: 000) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request Z Port (Port Z). 111: Makes the PCI Controller Master Z. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master Z. 010: Makes REQ*[2] Master Z. 001: Makes REQ*[1] Master Z. 000: Makes REQ*[0] Master Z.	R/W

Figure 10.4.21 PCI Bus Arbiter Request Port Register (2/2)

10.4.24 PCI Bus Arbiter Configuration Register (PBACFG) 0xD104

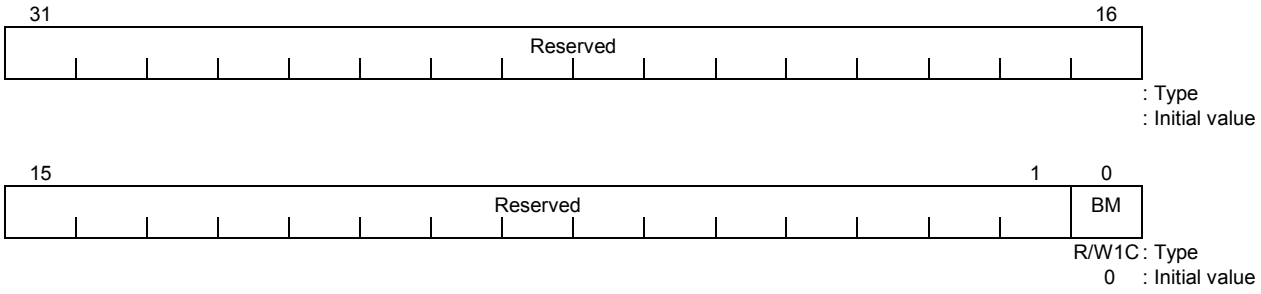
This register is only valid when using the on-chip PCI Bus Arbiter.

Bit	Mnemonic	Field Name	Description	Read/Write
31:4		Reserved		—
3	FIXPA	Fixed Park Master	Fixed Park Master (Default: 0) Selects the method for determining the Park Master. 0: The last Bus Master becomes the Park Master. 1: Internal PCI Bus Arbiter Request Port A is the Park Master.	R/W
2	RPBA	Reset PCI Bus Arbiter	Reset PCI Bus Arbiter (Default: 0) Resets the PCI Bus Arbiter. However, the PCI Bus Arbiter Register settings are saved. Please use the software to clear this bit. 1: The PCI Bus Arbiter is currently being reset. 0: The PCI Bus Arbiter is not currently being reset.	R/W
1	PBAEN	PCI Bus Arbiter Enable	PCI Bus Arbiter Enable (Default: 0) This is the Bus Arbiter Enable bit. After Reset, External PCI Bus requests to the PCI Arbiter cannot be accepted until this bit is set to “1”. The PCI Controller is the default Parking Master after Reset. 1: Enables the PCI Bus Arbiter. 0: Disables the PCI Bus Arbiter.	R/W
0	BMCEN	Broken Master Check Enable	Broken Master Check Enable (Default: 0) Controls Broken Master detection. 1: Enables the Broken PCI Bus Master check. 0: Disables the Broken PCI Bus Master check.	R/W

Figure 10.4.22 PCI Bus Arbiter Configuration Register

10.4.25 PCI Bus Arbiter Status Register (PBASTATUS) 0xD108

This register is only valid when using the on-chip PCI Bus Arbiter.

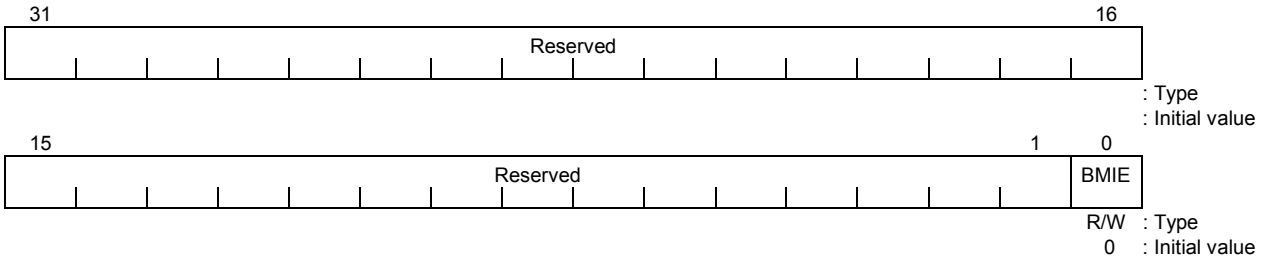


Bit	Mnemonic	Field Name	Description	Read/Write
31:1		Reserved		—
0	BM	Broken Master Detected	Broken Master Detected (Default: 0) This bit indicates that a Broken Master was detected. This bit is set to “1” if even one of the bits in the PCI Bus Arbiter Broken Master Register (PBABM) is “1”. 1: Indicates that a Broken Master was detected. 0: Indicates that no Broken Master has been detected.	R/W1C

Figure 10.4.23 PCI Bus Arbiter Status Register

10.4.26 PCI Bus Arbiter Interrupt Mask Register (PBAMASK) 0xD10C

This register is only valid when using the on-chip PCI Bus Arbiter.



Bit	Mnemonic	Field Name	Description	Read/Write
31:1		Reserved		—
0	BMIE	Broken Master Detected Interrupt Enable	Broken Master Detected Interrupt Enable (Default: 0) Generates an interrupt when a Broken Master is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W

Figure 10.4.24 PCI Bus Arbiter Interrupt Mask Register

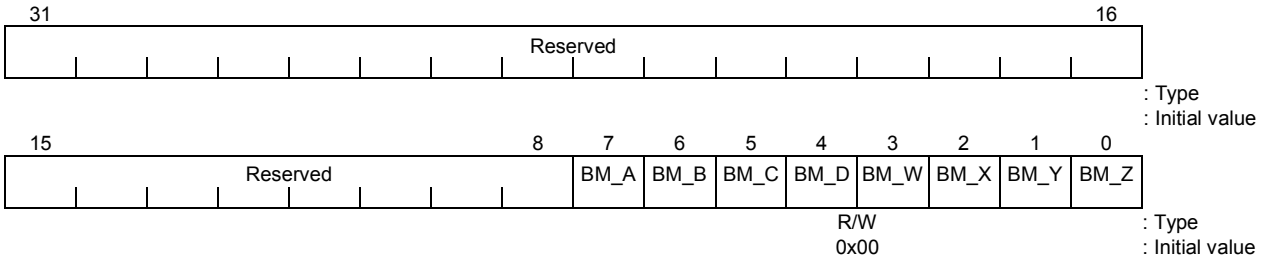
10.4.27 PCI Bus Arbiter Broken Master Register (PBABM) 0xD110

This register indicates the acknowledged Broken Master. This register sets the bit that corresponds to the PCI Master device that was acknowledged as the Broken Master when the Broken Master Check Enable bit (BMCEN) in the PCI Bus Arbiter Configuration Register (PBACFG) is set.

Regardless of the value of the Broken Master Check Enable bit, a PCI Master device is removed from the arbitration scheme when "1" is written to the corresponding BM bit.

This register must be cleared to "0" since bit mapping changes, making this register value invalid when the PCI Bus Arbiter Request Port Register (PBAREQPORT) is changed.

This register is only valid when using the on-chip PCI Bus Arbiter.

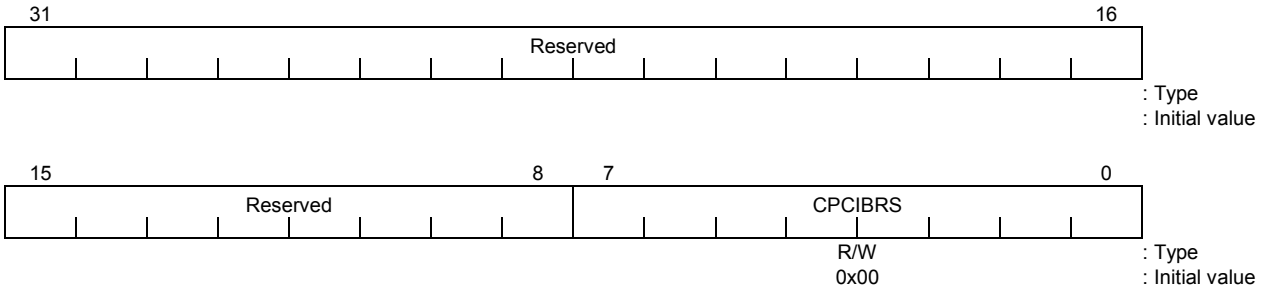


Bit	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7	BM_A	Broken Master	Broken Master A (Default: 0) Indicates whether PCI Bus Master A is a Broken Master. 1: PCI Bus Master A was acknowledged as a Broken Master. 0: PCI Bus Master A was not acknowledged as a Broken Master.	R/W
6	BM_B	Broken Master	Broken Master B (Default: 0) Indicates whether PCI Bus Master B is a Broken Master. 1: PCI Bus Master B was acknowledged as a Broken Master. 0: PCI Bus Master B was not acknowledged as a Broken Master.	R/W
5	BM_C	Broken Master	Broken Master C (Default: 0) Indicates whether PCI Bus Master C is a Broken Master. 1: PCI Bus Master C was acknowledged as a Broken Master. 0: PCI Bus Master C was not acknowledged as a Broken Master.	R/W
4	BM_D	Broken Master	Broken Master D (Default: 0) Indicates whether PCI Bus Master D is a Broken Master. 1: PCI Bus Master D was acknowledged as a Broken Master. 0: PCI Bus Master D was not acknowledged as a Broken Master.	R/W
3	BM_W	Broken Master	Broken Master W (Default: 0) Indicates whether PCI Bus Master W is a Broken Master. 1: PCI Bus Master W was acknowledged as a Broken Master. 0: PCI Bus Master W was not acknowledged as a Broken Master.	R/W
2	BM_X	Broken Master	Broken Master X (Default: 0) Indicates whether PCI Bus Master X is a Broken Master. 1: PCI Bus Master X was acknowledged as a Broken Master. 0: PCI Bus Master X was not acknowledged as a Broken Master.	R/W
1	BM_Y	Broken Master	Broken Master Y (Default: 0) Indicates whether PCI Bus Master Y is a Broken Master. 1: PCI Bus Master Y was acknowledged as a Broken Master. 0: PCI Bus Master Y was not acknowledged as a Broken Master.	R/W
0	BM_Z	Broken Master	Broken Master Z (Default: 0) Indicates whether PCI Bus Master Z is a Broken Master. 1: PCI Bus Master Z was acknowledged as a Broken Master. 0: PCI Bus Master Z was not acknowledged as a Broken Master.	R/W

Figure 10.4.25 PCI Bus Arbiter Broken Master Register

10.4.28 PCI Bus Arbiter Current Request Register (PBACREQ) 0xD114

This register is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.

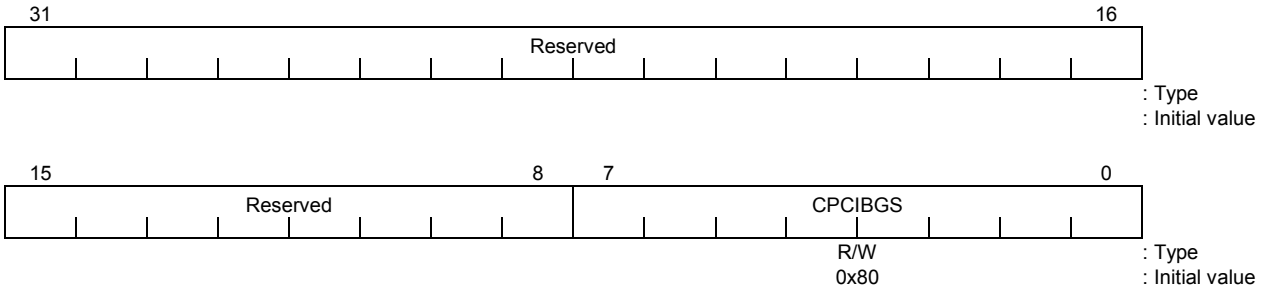


Bits	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7:0	CPCIBRS	Current PCI Bus Request Status	Current PCI Bus Request Status (Default: 0x00) This register indicates the status of the current PCI Bus Request Input Signal (PCI Controller and REQ*[3:0]). CPCIBRS[7] corresponds to the PCI Controller and CPCIBRS[3:0] correspond to REQ*[3:0].	R/W

Figure 10.4.26 PCI Bus Arbiter Current Request Register

10.4.29 PCI Bus Arbiter Current Grant Register (PBACGNT) 0xD118

This is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.

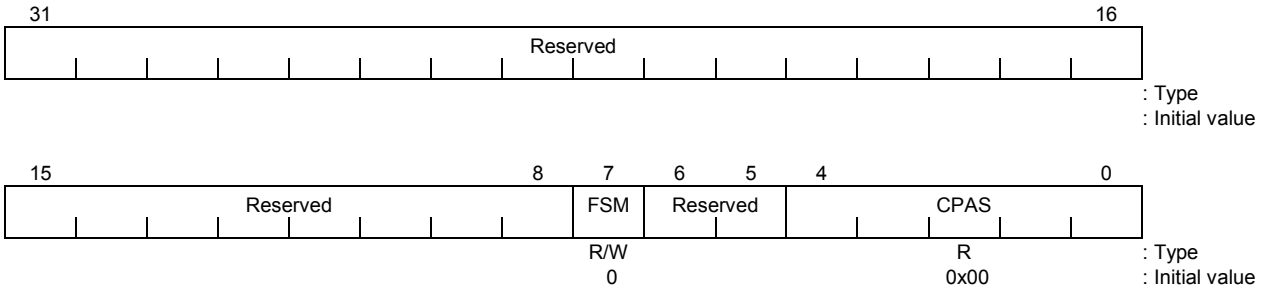


Bits	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7:0	CPCIBGS	Current PCI Grant Status	Current PCI Bus Grant Status (Default: 0x80) This register indicates the current PCI Bus Grant output signal (PCI Controller and GNT*[3:0]). CPCIBGS[7] corresponds to the PCI Controller, and CPCIBGS[3:0] correspond to GNT*[3:0].	R/W

Figure 10.4.27 PCI Bus Arbiter Current Grant Register

10.4.30 PCI Bus Arbiter Current State Register (PBACSTATE) 0xD11C

This is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.



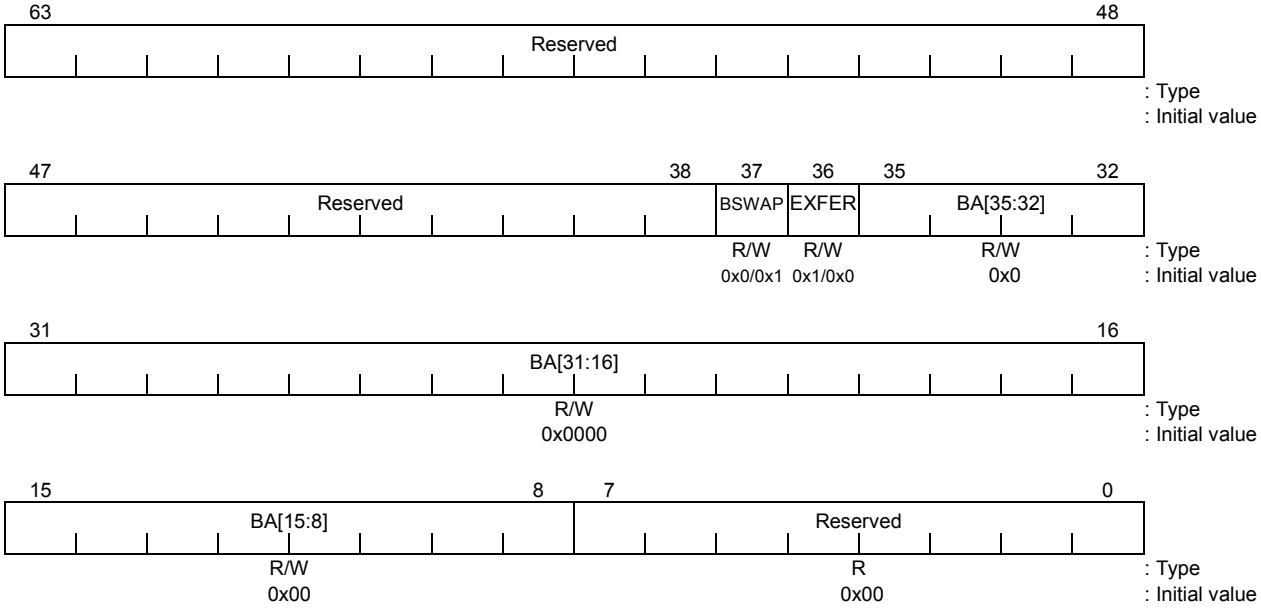
Bit	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7	FSM	Observe PCI Arbiter State Machine	Observe PCI Arbiter Finite State Machine (Default: 0) Specifies which State Machine to observe. 1: Observe the Level 1 State Machine. 0: Observe the Level 2 State Machine.	R/W
6:5		Reserved		—

Figure 10.4.28 PCI Bus Arbiter Current State Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
4:0	CPAS	Current PCI Bus Arbiter State	Current PCI Bus Arbiter State (Default: 0x00) Displays the State Machine that was selected by the FSM bit. Please refer to Figures 12.5.3 and 12.11.1 for an explanation of Agent/Grant A - W and Level 2. When FSM =1: 0x00: Preparation state for transferring bus ownership to PCI Agent A. 0x01: State in which Grant A is provided to PCI Agent A when PCI Bus ownership is being held elsewhere. 0x02: State in which Grant A is provided to PCI Agent A when PCI Bus ownership is not being held elsewhere. 0x03: The agent that was provided Grant A exists in this state. If there is bus ownership, the PCI Bus Arbiter transfers bus ownership to another agent. 0x04: Preparation state for transferring bus ownership to PCI Agent B. 0x05: State in which Grant B is provided to PCI Agent B when PCI Bus ownership is being held elsewhere. 0x06: State in which Grant B is provided to PCI Agent B when PCI Bus ownership is not being held elsewhere. 0x07: The agent that was provided Grant B exists in this state. If there is bus ownership, the PCI Bus Arbiter transfers bus ownership to another agent. 0x08: Preparation state for transferring bus ownership to PCI Agent C. 0x09: State in which Grant C is provided to PCI Agent C when PCI Bus ownership is being held elsewhere. 0x0A: State in which Grant C is provided to PCI Agent C when PCI Bus ownership is not being held elsewhere. 0x0B: The agent that was provided Grant C exists in this state. If there is bus ownership, the PCI Bus Arbiter transfers bus ownership to another agent. 0x0C: Preparation state for transferring bus ownership to PCI Agent D. 0x0D: State in which Grant D is provided to PCI Agent D when PCI Bus ownership is being held elsewhere. 0x0E: State in which Grant D is provided to PCI Agent D when PCI Bus ownership is not being held elsewhere. 0x0F: The agent that was provided Grant D exists in this state. If there is bus ownership, the PCI Bus Arbiter transfers bus ownership to another agent. 0x10: Preparation state for transferring bus ownership to PCI Agent Level 2. 0x11: State in which Grant Level 2 is provided to PCI Agent Level 2 when PCI Bus ownership is being held elsewhere. 0x12: State in which Grant Level 2 is provided to PCI Agent Level 2 when PCI Bus ownership is not being held elsewhere. 0x13: The agent that was provided Grant Level 2 exists in this state. If there is bus ownership, the PCI Bus Arbiter transfers bus ownership to another agent. When FSM=0, the FSM=1 description is replaced as follows: A→W, B→X, C→Y, D→Z, Level 2→N/A.	R

Figure 10.4.28 PCI Bus Arbiter Current State Register (2/2)

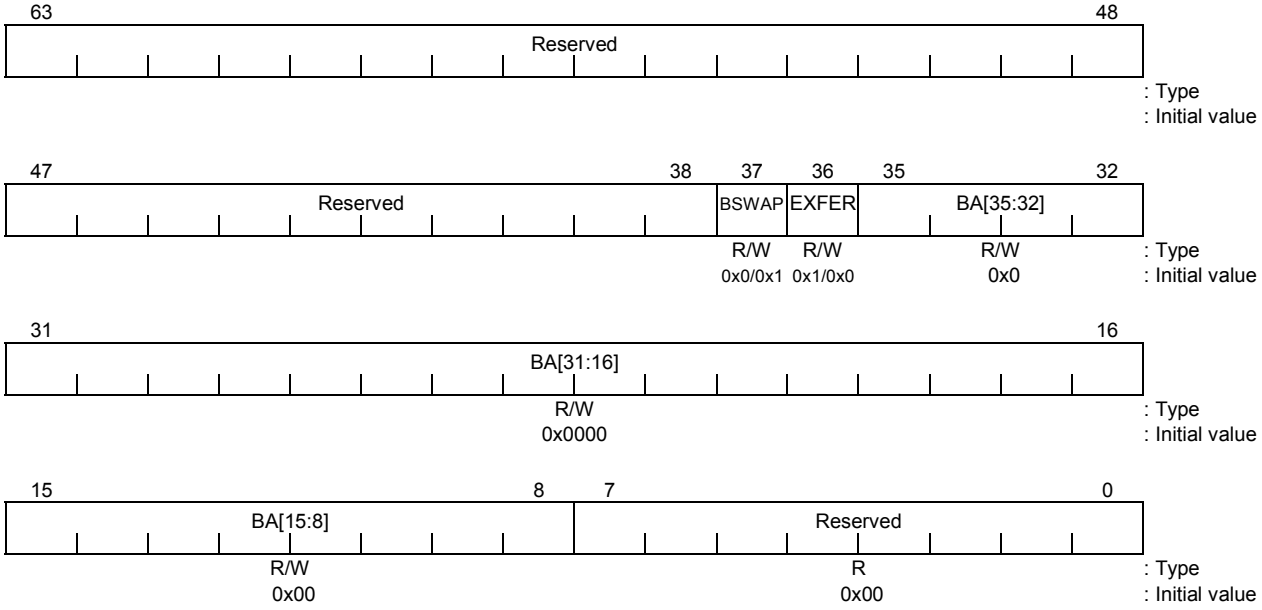
10.4.31 G2P Memory Space 0 G-Bus Base Address Register (G2PM0GBASE) 0xD120



Bit	Mnemonic	Field Name	Description	Read/Write
63:38		Reserved		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 0. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 0. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:8	BA[35:8]	Base Address	Base Address (Default: 0x0_0000_00) Sets the G-Bus base bus address of Memory Space 0 for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Reserved		R

Figure 10.4.29 G2P Memory Space 0 G-Bus Base Address Register

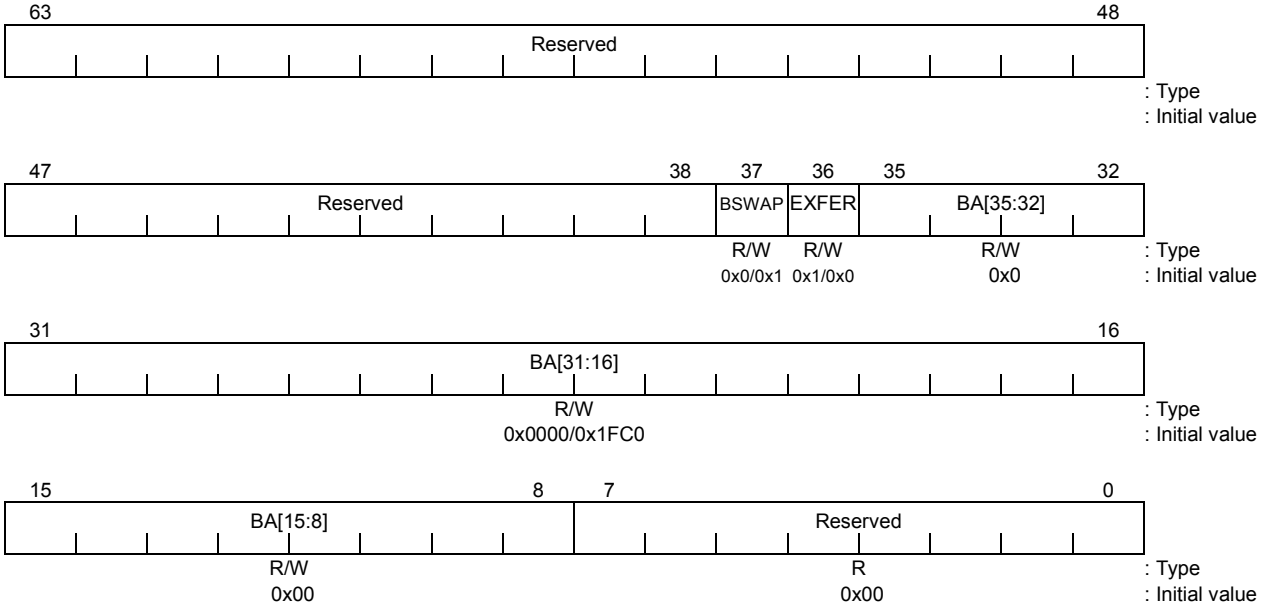
10.4.32 G2P Memory Space 1 G-Bus Base Address Register (G2PM1GBASE) 0xD128



Bit	Mnemonic	Field Name	Description	Read/Write
63:38		Reserved		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 1. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 1. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:8	BA[35:8]	Memory Space Base Address 1	Base Address (Default: 0x0_0000_00) Sets the G-Bus base bus address of Memory Space 1 for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Reserved		R

Figure 10.4.30 G2P Memory Space 1 G-Bus Base Address Register

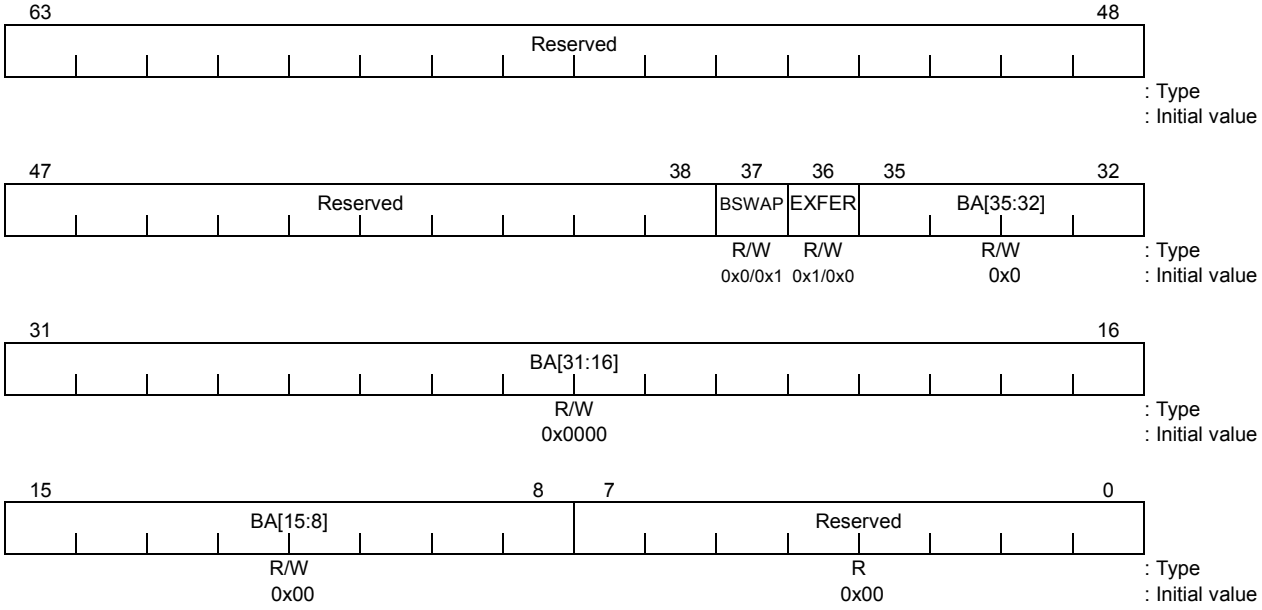
10.4.33 G2P Memory Space 2 G-Bus Base Address Register (G2PM2GBASE) 0xD130



Bit	Mnemonic	Field Name	Description	Read/Write
63:38		Reserved		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 0. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 0. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:8	BA[35:8]	Base Address	Base Address (Default: Normal Mode: 0x0_0000_00; PCI Boot Mode: 0x0_1FC0_00) Sets the G-Bus base bus address of Memory Space 2 for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Reserved		R

Figure 10.4.31 G2P Memory Space 2 G-Bus Base Address Register

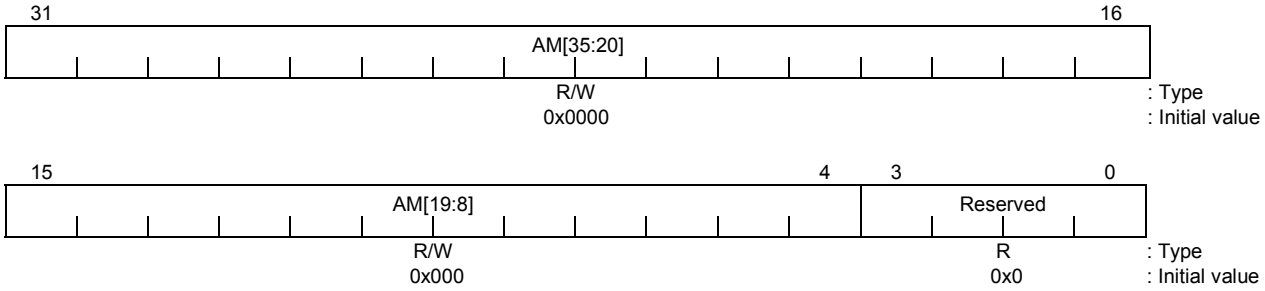
10.4.34 G2P I/O Space G-Bus Base Address Register (G2PIOGBASE) 0xD138



Bit	Mnemonic	Field Name	Description	Read/Write
63:38		Reserved		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of the I/O space. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to "1" when in the Big Endian Mode, the byte order of transfer to the I/O Memory Space through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of the I/O Space. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:8	BA[35:8]	Base Address	Base Address (Default: 0x0_0000_00) Sets the G-Bus base bus address of the I/O Memory Space for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Reserved		R

Figure 10.4.32 G2P I/O Space G-Bus Address Register

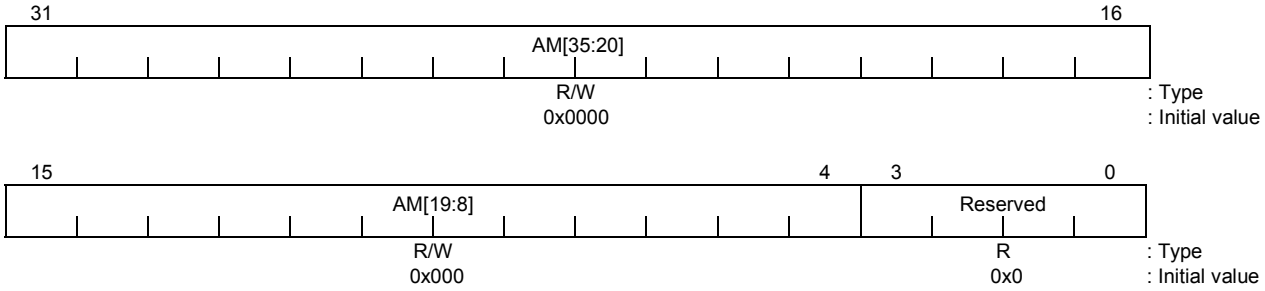
10.4.35 G2P Memory Space 0 Address Mask Register (G2PM0MASK) 0xD140



Bit	Mnemonic	Field Name	Description	Read/Write
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Default: 0x0_0000_00) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x00FF_FFF0.	R/W
3:0		Reserved		R

Figure 10.4.33 G2P Memory Space 0 Address Mask Register

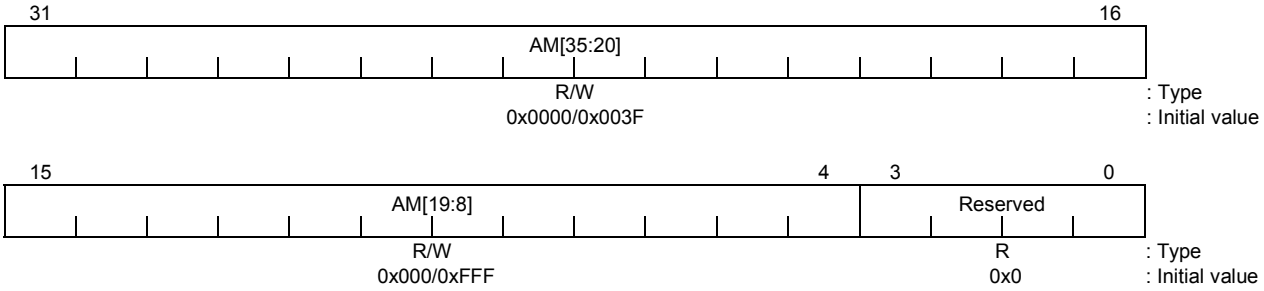
10.4.36 G2P Memory Space 1 Address Mask Register (G2PM1MASK) 0xD144



Bit	Mnemonic	Field Name	Description	Read/Write
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Default: 0x0_0000_00) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x00FF_FFF0.	R/W
3:0		Reserved		R

Figure 10.4.34 G2P Memory Space 1 Address Mask Register

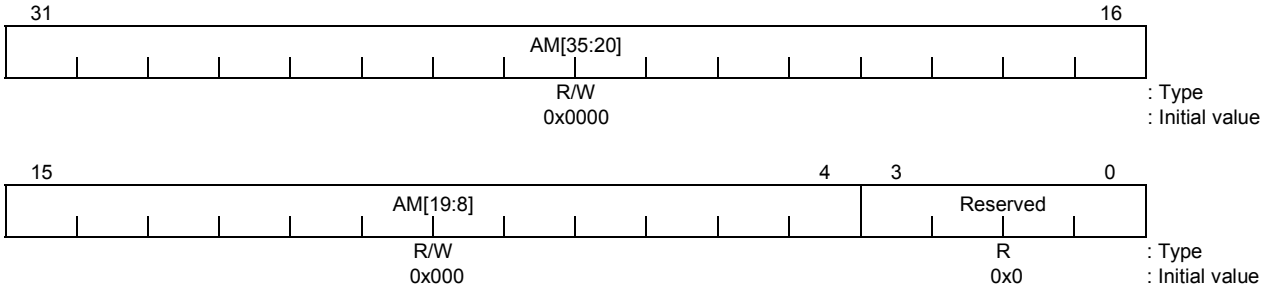
10.4.37 G2P Memory Space 2 Address Mask Register (G2PM2MASK) 0xD148



Bit	Mnemonic	Field Name	Description	Read/Write
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Default: 0x0_0000_00) (Default: Normal Mode: 0x0_0000_00; PCI Boot Mode: 0x0_03FF_FF) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x00FF_FFF0. Note: To boot PCI, set 0x0_003F_FF (4 Mbyte space) to AM[35:8] in the boot code.	R/W
3:0		Reserved		R

Figure 10.4.35 G2P Memory Space 2 Address Mask Register

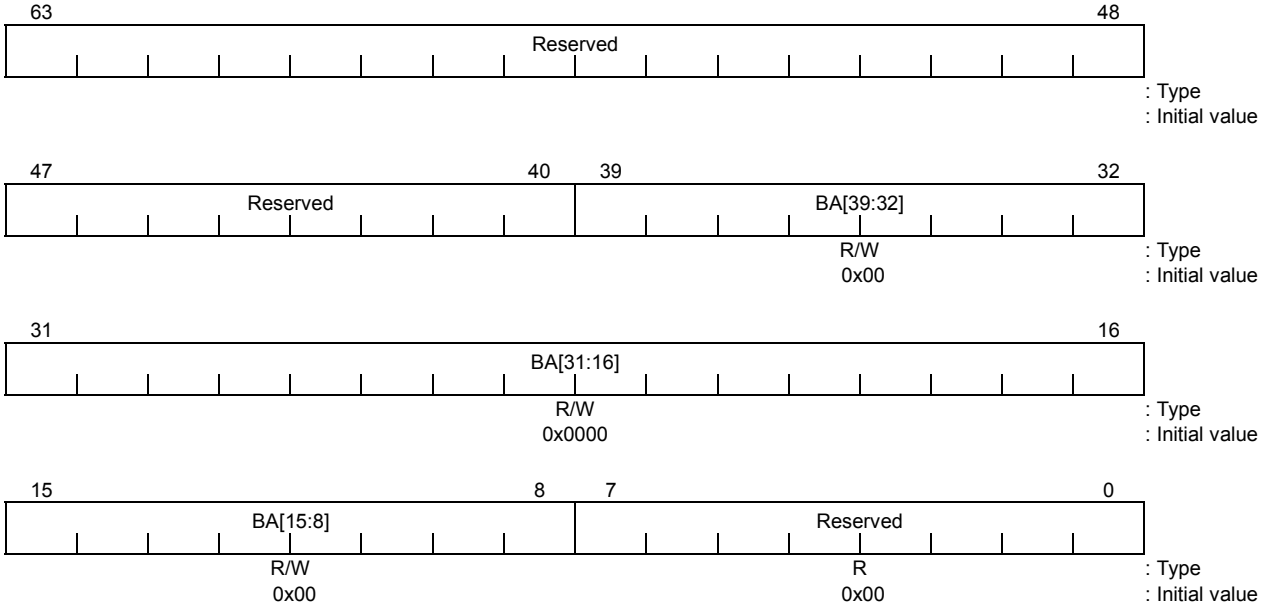
10.4.38 G2P I/O Space Address Mask Register (G2PIOMASK) 0xD14C



Bits	Mnemonic	Field Name	Description	Read/Write
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Default: 0x0_0000_00) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x0000_0100) for example, the value becomes 0x0000_0000.	R/W
3:0		Reserved		R

Figure 10.4.36 G2P I/O Space Address Mask Register

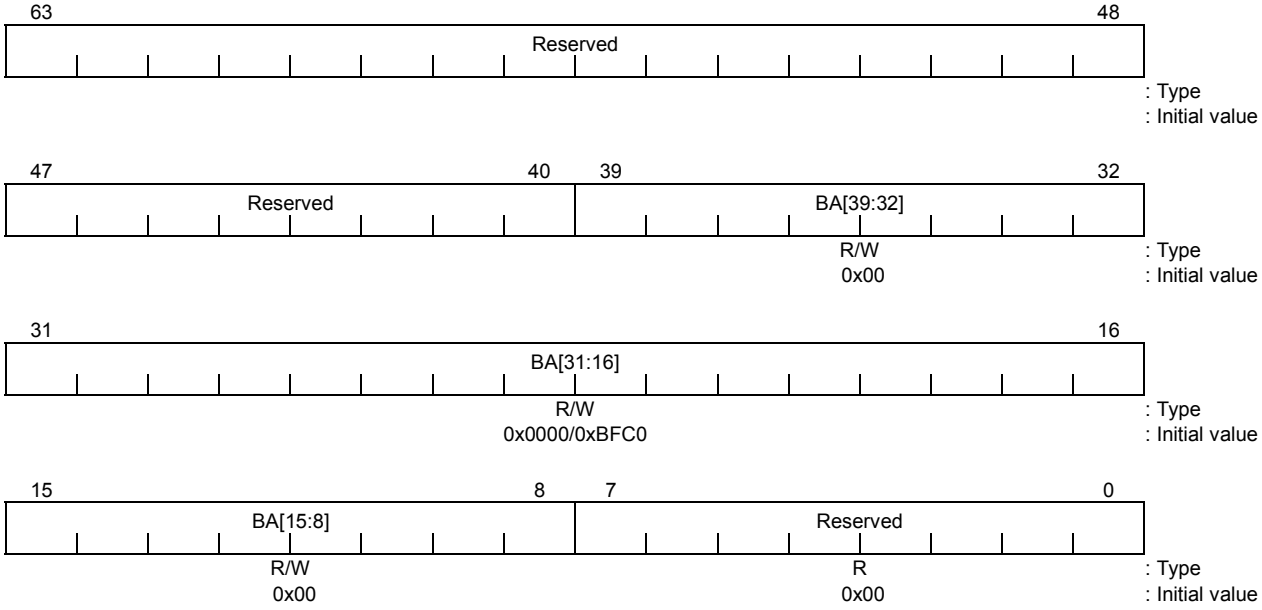
10.4.39 G2P Memory Space 0 PCI Base Address Register (G2PM0PBASE) 0xD150



Bits	Mnemonic	Field Name	Register	Read/Write
63:40		Reserved		—
39:8	BA[39:8]	Base Address	Base Address (Default: 0x00_0000_00) Sets the PCI Base address of Memory Space 0 for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Reserved		R

Figure 10.4.37 G2P Memory Space 0 G-Bus Base Address Register

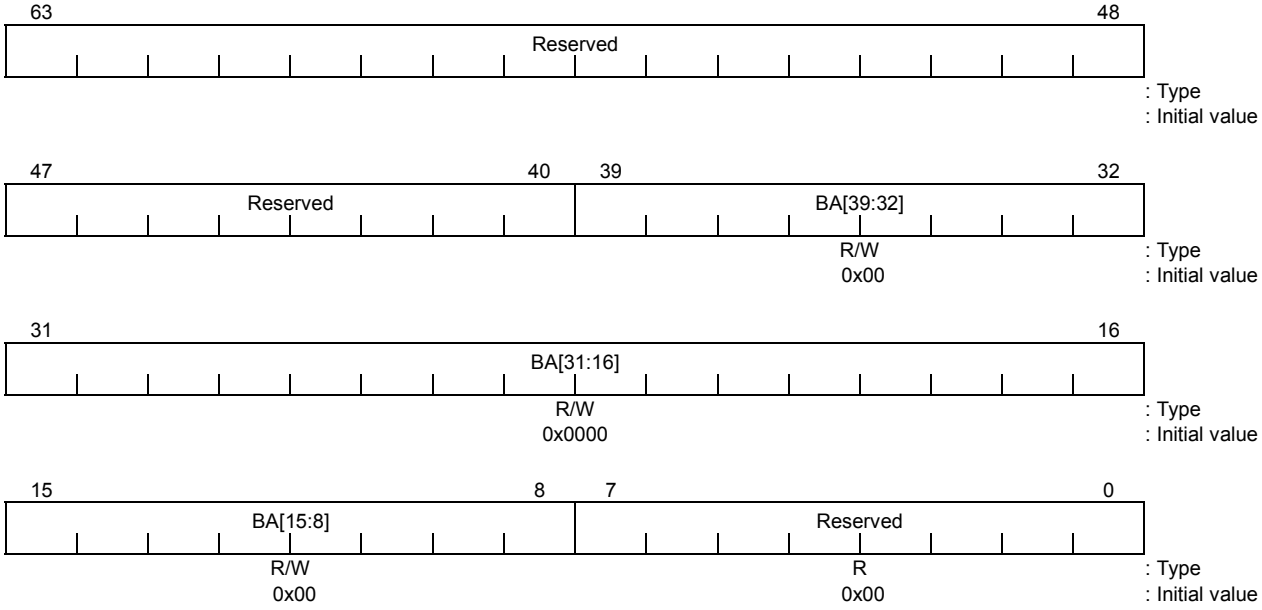
10.4.40 G2P Memory Space 1 PCI Base Address Register (G2PM1PBASE) 0xD158



Bit	Mnemonic	Field Name	Description	Read/Write
63:40		Reserved		—
39:8	BA[39:8]	Base Address	Base Address (Default: 0x00_0000_00) Sets the PCI Base address of Memory Space 1 for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Reserved		R

Figure 10.4.38 G2P Memory Space 1 G-Bus Base Address Register

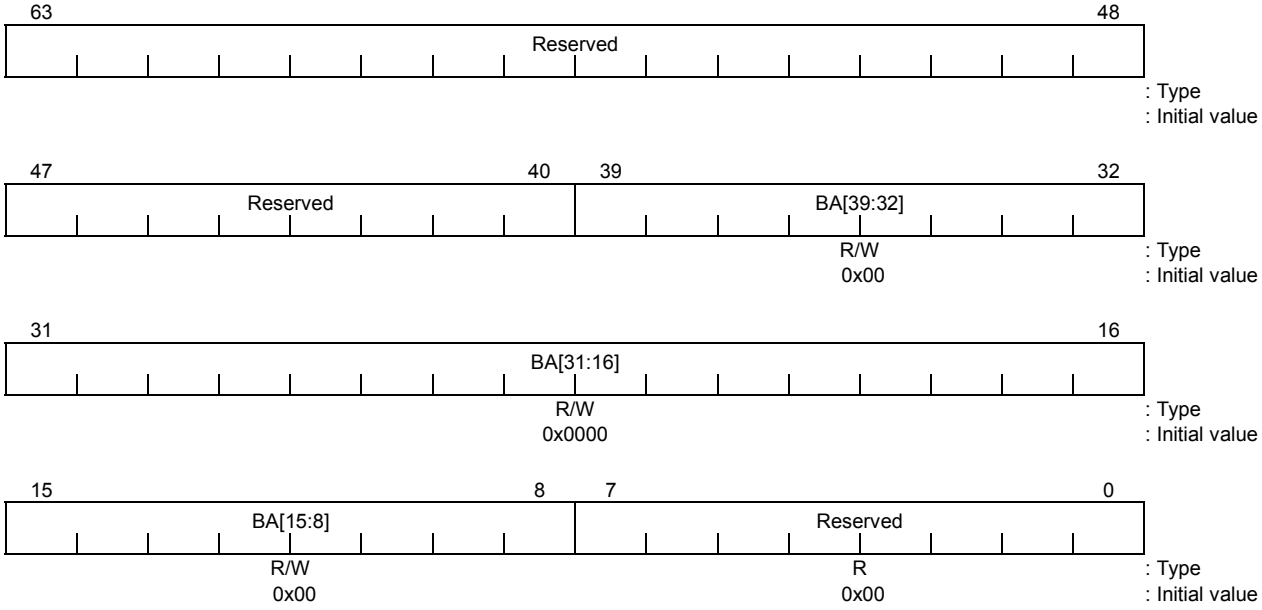
10.4.41 G2P Memory Space 2 PCI Base Address Register (G2PM2PBASE) 0xD160



Bits	Mnemonic	Field Name	Description	Read/Write
63:40		Reserved		—
39:8	BA[39:8]	Base Address	Base Address (Default: 0x00_0000_00) Sets the PCI Base address of Memory Space 2 for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Reserved		R

Figure 10.4.39 G2P Memory Space 2 G-Bus Base Address Register

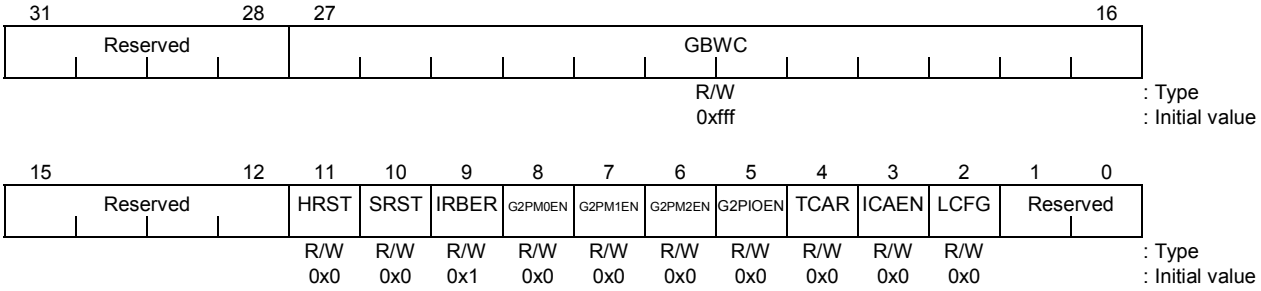
10.4.42 G2P I/O Space PCI Base Address Register (G2PIOPBASE) 0xD168



Bits	Mnemonic	Field Name	Description	Read/Write
63:40		Reserved		—
39:8	BA[39:8]	Base Address	Base Address (Default: 0x00_0000_00) Sets the PCI Base address of the I/O Space for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Reserved		R

Figure 10.4.40 G2P I/O Space G-Bus Address Register

10.4.43 PCI Controller Configuration Register (PCICCFG) 0xD170



Bit	Mnemonic	Field Name	Description	Read/Write
31:28		Reserved		—
27:16	GBWC	G-Bus Wait Counter Setting	G Bus Wait Counter (Default: 0xFFF) Sets the Retry response counter at the G-Bus during a PCI initiator Read transaction. When the initiator Read access cycle exceeds the setting of this counter, a Retry response is sent to the G-Bus and the G-Bus is released. PCI Read operation continues. This counter uses the G-Bus clock (GBUSCLK) when operating. When 0x000 is set, a Retry response is not sent to the G-Bus by a long response cycle count. When the G-Bus timeout count is used with the value other than the initial value 4096 GBUSCLK, G-BUS timeout may occur before a Retry response is sent. When G-Bus timeout of the configuration register (CCFG.GTOT) is used with the value other than the initial value (11), set the following maximum values to the register. GTOT value Maximum value of the register 10 (2048 GBUSCLK) : 0x7f0 01 (1024 GBUSCLK) : 0x3f0 00 (512 GBUSCLK) : 0x1f0	R/W
15:12		Reserved		—
11	HRST	Hardware Reset	Hard Reset (Default: 0x0) Performs PCI Controller hardware reset control. EEPROM reloading is also performed. This bit is automatically cleared when Reset ends. This is a diagnostic function. The PCI Controller cannot be accessed for 32 G-Bus clock cycles after this bit is set. 1: Perform a hardware reset on the PCI Controller. 0: Do not perform a hardware reset on the PCI Controller.	R/W

Figure 10.4.41 PCI Controller Configuration Register (1/3)

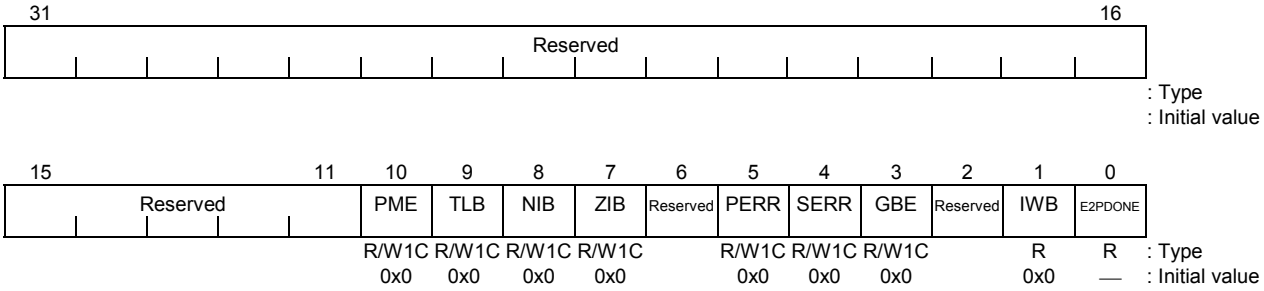
Bit	Mnemonic	Field Name	Description	Read/Write
10	SRST	Software Reset	Soft Reset (Default: 0x0) Performs PCI Controller software reset control. Data is also reloaded to the Configuration Space Register from EEPROM or from the Configuration Data Register. Please set this bit after the EEPROM Load End bit (PCICSTATUS.E2PDONE) is set. Also, please use the software to clear this bit at least four PCI Bus Clock cycles after Reset. Other registers of the PCI Controller cannot be accessed while this bit is set. This bit differs from the Hardware Reset bit (HRST). The following register values are not initialized. • G2P Status Register (G2PSTATUS) • PCI Bus Arbiter Status Register (PBASTATUS) • PCI Controller Status Register (PCICSTATUS) • Software Reset bit (PCICCFG.SRST) • Load Configuration Register bit (PCICCFG.LCFG) 1: The PCI Controller is reset by the software. 0: The PCI Controller is not reset by the software.	R/W
9	IRBER	Bus Error Response Setting During Initiator Read	Initiator Read Bus Error Response (Default: 0x1) Bus error responses on the G-Bus are controlled when the following phenomena indicated by the PCI Status, Command Register (PICSTATUS) and the G2P Status Register (G2PSTATUS) occur during initiator Read access. Detected Parity Error (PCISTATUS.DPE) Received Master Abort (PCISTATUS.RMA) Received Target Abort (PCISTATUS.RTA) Initiator Detected TRDY Time Out Error (G2PSTATUS.IDTTOE) Initiator Detected Retry Time Out Error (G2PSTATUS.IDRTOE) 1: Responds with a Bus error on the G-Bus. 0: Does not respond with a Bus error on the G-Bus. (Normally terminates the Read transaction on the G-Bus. Read data is invalid.)	R/W
8	G2PM0EN	Initiator Memory Space 0 Enable	Initiator Memory Space 0 Enable (Default: 0x0) Controls PCI initiator access to Memory Space 0. 1: Memory Space 0 is valid. 0: Memory Space 0 is invalid.	R/W
7	G2PM1EN	Initiator Memory Space 1 Enable	Initiator Memory Space 1 Enable (Default: 0x0) Controls PCI initiator access to Memory Space 1. 1: Memory Space 1 is valid. 0: Memory Space 1 is invalid.	R/W
6	G2PM2EN	Initiator Memory Space 2 Enable	Initiator Memory Space 2 Enable (Default: Normal Mode: 0x0; PCI Boot Mode: 0x1) Controls PCI initiator access to Memory Space 2. 1: Memory Space 2 is valid. 0: Memory Space 2 is invalid.	R/W
5	G2PIOEN	Initiator I/O Space Enable	Initiator I/O Space Enable (Default: 0x0) Controls PCI initiator access to the I/O Space.. 1: I/O Space is valid. 0: I/O Space is invalid.	R/W

Figure 10.4.41 PCI Controller Configuration Register (2/3)

Bit	Mnemonic	Field Name	Description	Read/Write
4	TCAR	Target Configuration Access Ready	Target Configuration Access Ready (Default: 0x0/0x1) Specifies whether to accept PCI access as a target. PCI controller receives a target access, when this bit is 1 and PCISTATUS.E2PDONE bit is 1. Configuration access from the PCI Bus can be accepted during PCI Boot up after initialization from EEPROM or after each initialization ends. Please use the software to set this bit after initialization ends. Retry response to PCI configuration access is performed until this bit is set. This bit becomes “1” only when in the PCI Boot Mode and the Satellite Mode. Operation when this bit is set to “1” then reset to “0” is not defined. 1: Responds to PCI target access. 0: Performs a Retry response to PCI target access.	R/W
3	ICAEN	Initiator Configuration Access Enable	Initiator Configuration Access Enable (Default: 0x1) Controls initiator PCI configuration access using the G2P Configuration Address Register (G2PCFGADRS) and the G2P Configuration Data Register (G2PCFGDATA). This is a diagnostic function. 1: Initiator configuration access is possible. 0: Initiator configuration access is not possible.	R/W
2	LCFG	Load Configuration Data Register	Load PCI Configuration Data Register (Default: 0x0) When a software reset is performed on this bit using the Software Reset bit (PCICFG.SRST) when this bit is already set, data is loaded to the Configuration Space Register from the Configuration Data 0/1/2/3 Register. 1: Load from the Configuration Data 0/1/2/3 Register. 0: Load from EEPROM.	R/W
1:0		Reserved		—

Figure 10.4.41 PCI Controller Configuration Register (3/3)

10.4.44 PCI Controller Status Register (PCICSTATUS) 0xD174



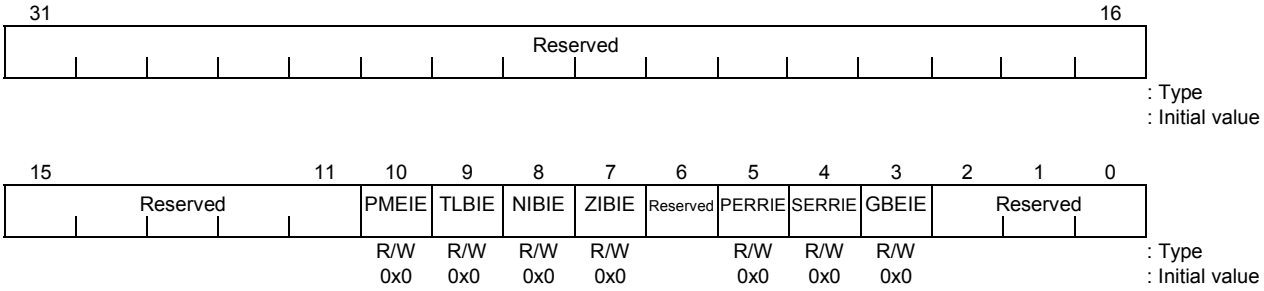
Bit	Mnemonic	Field Name	Description	Read/Write
31:11		Reserved		—
10	PME	PME Detect	PME Detect (Default: 0x0) When the PCI Controller is in the Host mode, this bit indicates that assertion of the PME* signal was detected. 1: Indicates that assertion of the PME* signal was detected. 0: Indicates that assertion of the PME* signal was not detected.	R/W1C
9	TLB	Long Burst Transfer Detect	Too Long Burst Detect (Default: 0x0) Indicates that a Burst transfer by the on-chip DMA Controller exceeding 8 DWORDs was detected. 1: Indicates that a Burst transfer exceeding 8 DWORDs was detected. 0: Indicates that no Burst transfer exceeding 8 DWORDs was detected.	R/W1C
8	NIB	Negative Increment Burst Detect	Negative Increment Burst Detect (Default: 0x0) Indicates that Burst transfer by the on-chip DMA Controller in the negative direction was detected. 1: Indicates that a Burst transfer in the negative direction was detected. 0: Indicates that no Burst transfer in the negative direction was detected.	R/W1C
7	ZIB	Zero Increment Burst Detect	Zero Increment Burst Detect (Default: 0x0) Indicates that Burst transfer by the on-chip DMA Controller without an address increment was detected. 1: Indicates that a Burst transfer without an address increment was detected. 0: Indicates that no Burst transfer without an address increment was detected.	R/W1C
6		Reserved		—
5	PERR	PERR* Detected	PERR* Occurred (Default: 0x0) Indicates that the Parity Error signal (PERR*) was asserted. This bit is a monitor status bit that records assertion of the PERR* signal even if the TX4938 is not accessing PCI. 1: Indicates that the PERR* signal was asserted. 0: Indicates that the PERR* signal was not asserted.	R/W1C
4	SERR	SERR* Detected	SERR* Occurred (Default: 0x0) Indicates that the System Error signal (SERR*) was asserted. This bit is a monitor status bit that records assertion of the SERR* signal even if the TX4938 is not accessing PCI. 1: Indicates that the SERR* signal was asserted. 0: Indicates that the SERR* signal was not asserted.	R/W1C
3	GBE	G-Bus Error Detect	G-Bus Error Detect (Default: 0x0) Indicates that a G-Bus Error occurred in the G-Bus Master cycle of the PCI Controller. This error is indicated when a timeout occurs on the G-Bus. This bit is only set by Master cycle Bus Errors. 1: Indicates that a G-Bus Error was detected. 0: Indicates that no G-Bus Error was detected.	R/W1C

Figure 10.4.42 PCI Controller Status Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
2		Reserved		—
1	IWB	Initiator Write Busy	Initiator Write Busy (Busy: 0x0) Indicates that a Write cycle was in progress when a Write cycle to the PCI Bus was executed. While a Write cycle is in progress, no error status to that Write cycle is reflected. Therefore, this bit is used to confirm the status when it changes from “1” to “0” after the Write cycle ends. 1: Indicates that a Write cycle is in progress. 0: Indicates that no Write cycle is in progress.	R
0	E2PDONE	EEPROM Load Done	EEPROM Load Done (Default--) When using EEPROM, this bit indicates that data loading from EEPROM is complete. This bit is set to “1” when the internal process ends even if no EEPROM is connected. 1: Indicates that data loading from EEPROM is complete. 0: Indicates that data loading from EEPROM is not complete.	R

Figure 10.4.42 PCI Controller Status Register (2/2)

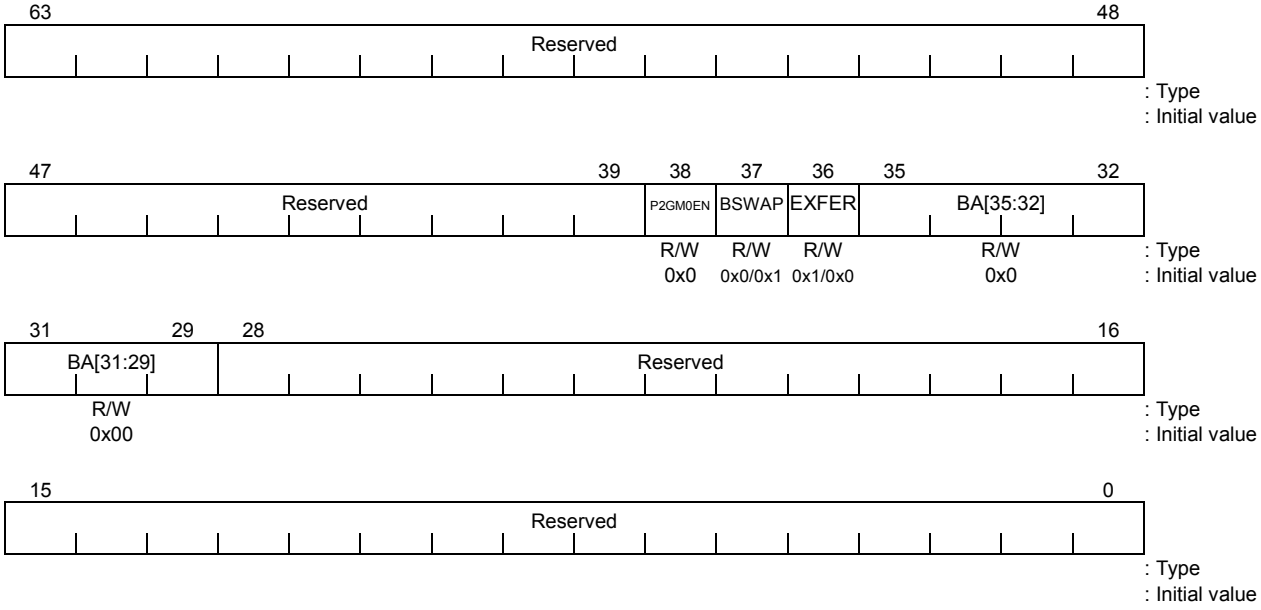
10.4.45 PCI Controller Interrupt Mask Register (PCICMASK) 0xD178



Bit	Mnemonic	Field Name	Description	Read/Write
31:11		Reserved		—
10	PMEIE	PME Detect Interrupt Enable	PME* Signal Interrupt Enable (Default: 0x0) When in the Host mode, this bit generates an interrupt when input of the PME* signal is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
9	TLBIE	Long Burst Transfer Detect Interrupt	Too Long Burst Interrupt Enable (Default: 0x0) This bit generates an interrupt when a Burst transfer by the on-chip DMA Controller exceeding 8 DWORDs was detected. 1: Generates an interrupt. 0: Does not generate an interrupt	R/W
8	NIBIE	Negative Increment Burst Transfer Detect Interrupt Enable	Negative Increment Burst Interrupt Enable (Default: 0x0) This bit generates an interrupt when a negative direction Burst transfer by the on-chip DMA Controller is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
7	ZIBIE	Zero Increment Burst Transfer Detect Interrupt Enable	Zero Increment Burst Interrupt Enable (Default: 0x0) This bit generates an interrupt when a Burst transfer by the on-chip DMA Controller without an address increment is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
6		Reserved		—
5	PERRIE	PERR* Detect Interrupt Enable	PERR* Interrupt Enable (Default: 0x0) This bit generates an interrupt when the Parity Error signal (PERR*) is asserted. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
4	SERRIE	SERR* Detect Interrupt Enable	SERR* Interrupt Enable (Default: 0x0) This bit generates an interrupt when the System Error signal (SERR*) is asserted. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
3	GBEIE	G-Bus Bus Error Detect Interrupt Enable	G-Bus Bus Error Interrupt Enable (Default: 0x0) This bit generates an interrupt when a Bus Error is asserted while the PCI Controller is the G-Bus Master. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
2:0		Reserved		—

Figure 10.4.43 PCI Controller Interrupt Mask Register

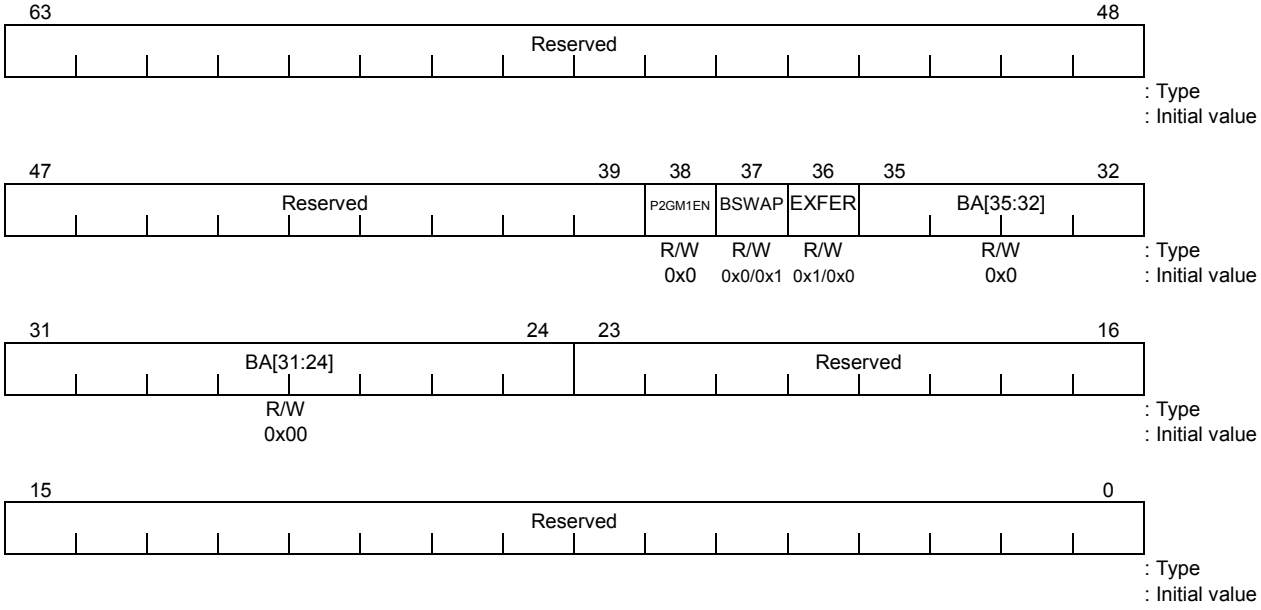
10.4.46 P2G Memory Space 0 G-Bus Base Address Register (P2GM0GBASE) 0xD180



Bit	Mnemonic	Field Name	Description	Read/Write
63:39		Reserved		—
38	P2GM0EN	Memory Space 0 Enable	Target Memory Space 0 Enable (Default: 0x0) Controls whether Memory Space 0 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 0 Lower Base Address Register or the Memory Space 0 Upper Base Address Register of the PCI Configuration Register become invalid. Also, “0” is returned to Reads as a response. 1: Validates Memory Space 0 for target access. 0: Invalidates Memory Space 0 for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 0 for target access.. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 0 for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:29	BA[35:29]	Base Address	Base Address 0 (Default: 0x000) Sets the G-Bus base bus address of Memory Space 0 for target access. Can set the base address in 512-MB units.	R/W
28:0		Reserved		—

Figure 10.4.44 P2G Memory Space 0 G-Bus Base Address Register

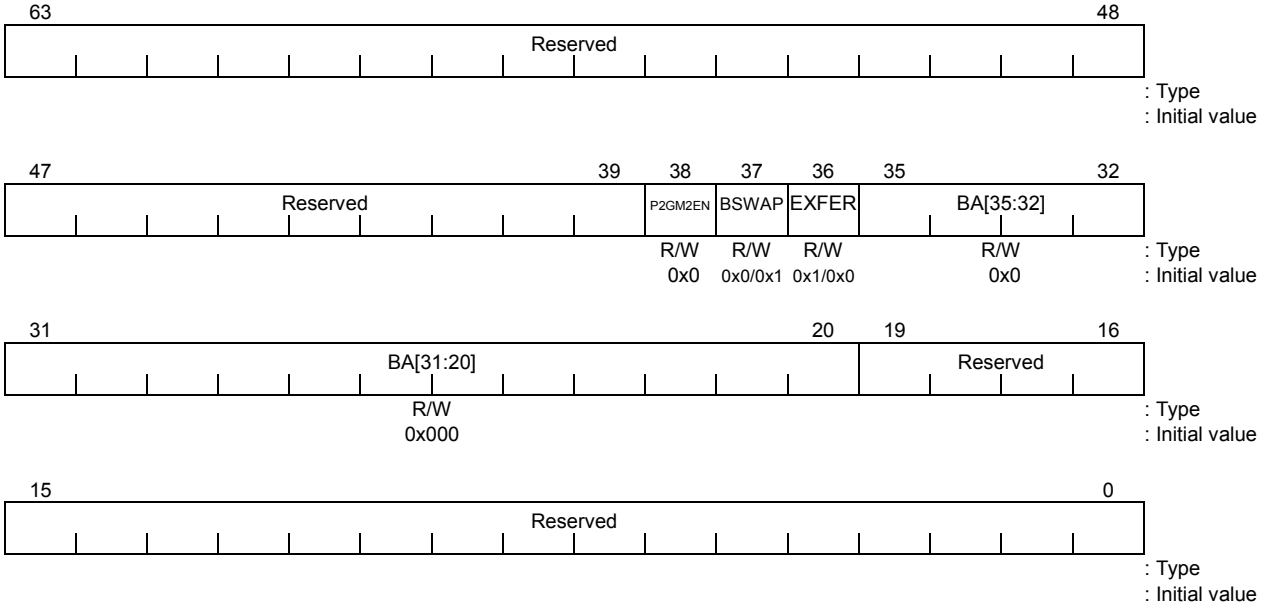
10.4.47 P2G Memory Space 1 G-Bus Base Address Register (P2GM1GBASE) 0xD188



Bit	Mnemonic	Field Name	Description	Read/Write
63:39		Reserved		—
38	P2GM1EN	Memory Space 1 Enable	Target Memory Space 1 Enable (Default: 0x0) Controls whether Memory Space 1 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 1 Lower Base Address Register or the Memory Space 1 Upper Base Address Register of the PCI Configuration Register become invalid. Also, “1” is returned to Reads as a response. 1: Validates Memory Space 1 for target access. 0: Invalidates Memory Space 1 for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 1 for target access. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 1 for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:24	BA[35:24]	Memory Space Base Address 1	Base Address 0 (Default: 0x0_0000_00) Sets the G-Bus base bus address of Memory Space 1 for target access. Can set the base address in 16-MB units.	R/W
23:0		Reserved		—

Figure 10.4.45 P2G Memory Space 1 G-Bus Base Address Register

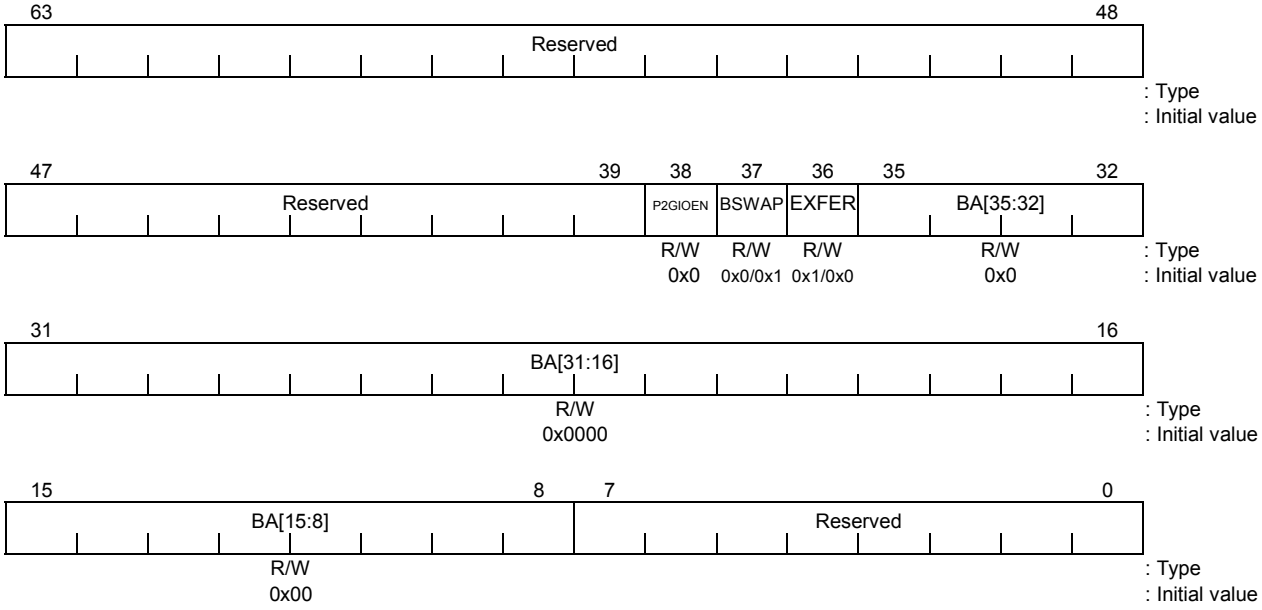
10.4.48 P2G Memory Space 2 G-Bus Base Address Register (P2GM2GBASE) 0xD190



Bit	Mnemonic	Field Name	Description	Read/Write
63:39		Reserved		—
38	P2GM2EN	Memory Space 2 Enable	Target Memory Space 2 Enable (Default: 0x0) Controls whether Memory Space 2 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 2 Lower Base Address Register or the Memory Space 2 Upper Base Address Register of the PCI Configuration Register become invalid. Also, “0” is returned to Reads as a response. 1: Validates Memory Space 2 for target access. 0: Invalidates Memory Space 2 for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 2 for target access. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 2 through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 2 for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:20	BA[35:20]	Memory Space Base Address 2	Base Address 2 (Default: 0x000) Sets the G-Bus base bus address of Memory Space 2 for target access. Can set the base address in 1-MB units.	R/W
19:0		Reserved		—

Figure 10.4.46 P2G Memory Space 2 G-Bus Base Address Register

10.4.49 P2G I/O Space G-Bus Base Address Register (P2GIOGBASE) 0xD198

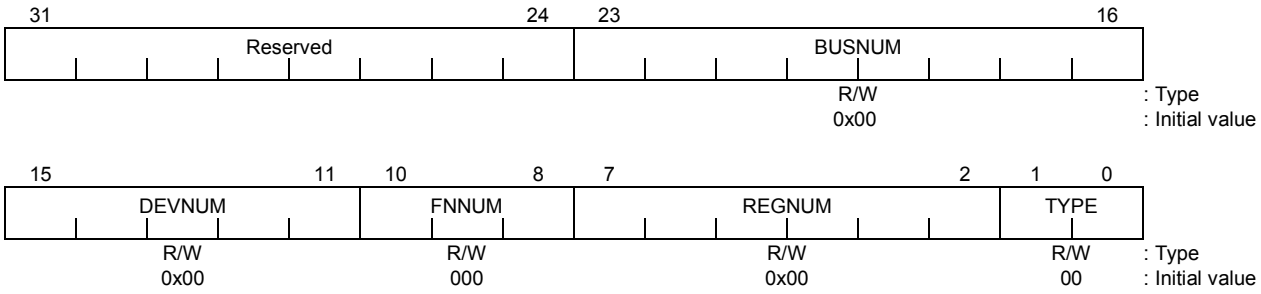


Bit	Mnemonic	Field Name	Description	Read/Write
63:39		Reserved		—
38	P2GIOEN	I/O Space Enable	Target I/O Space Enable (Default: 0x0) Controls whether the I/O Space for target access is valid or invalid. When this bit is set to invalid, Writes to the I/O Space Base Address Register of the PCI Configuration Register become invalid. Also, “0” is returned to Reads as a response. 1: Validates I/O Space for target access. 0: Invalidates I/O Space for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of the I/O Space for target access. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to the I/O Space through DWORD (32-bit) access will not change.	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of the I/O Space for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state.	R/W
35:8	BA[35:8]	Memory Space Base Address 2	Base Address 2 (Default: 0x000) Sets the G-Bus base bus address of the I/O Space for target access. Can set the base address in 256-byte units.	R/W
7:1		Reserved		—

Figure 10.4.47 P2G I/O Space G-Bus Base Address Register

10.4.50 G2P Configuration Address Register(G2PCFGADRS) 0xD1A0

The operation of any access to this register is undefined when the PCI Controller is in the Satellite mode.



Bit	Mnemonic	Field Name	Description	Read/Write
31:24		Reserved		—
23:16	BUSNUM	Bus Number	Bus Number (Default: 0x00) Indicates the target PCI Bus Number (one of 256).	R/W
15:11	DEVNUM	Device Number	Device Number (Default: 0x00) This field is used to identify the target physical device number. (This is one number out of 32 devices. 21 of these 32 devices are used.) When in the address phase of Type 0 configuration access, AD[31:11] of the upper 21 address lines are used as the IDSEL signal. 0x00: Use AD [11] as IDSEL. 0x01: Use AD [12] as IDSEL. 0x02: Use AD [13] as IDSEL. : : 0x13: Use AD [30] as IDSEL. 0x14: Use AD [31] as IDSEL. 0x15 - 0x1F: Reserved	R/W
10:8	FNNUM	Function Number	Function Number (Default: 000) This field is used to identify the target logic function number (one out of 8).	R/W
7:2	REGNUM	Register Number	Register Number (Default: 0x00) This field is used to identify the DWORD (one out of 64) inside the Configuration Space of the target function	R/W
1:0	TYPE	Type	Type (Default; 00) This field is used to identify the address type in the address phase of the target function configuration cycle. 0x0: Type 0 configuration (Use the AD[31:11] signal as the IDSEL signal.) 0x1: Type 1 configuration (Output all bits unchanged as the address to the AD[ ] signal.)	R/W

Figure 10.4.48 G2P Configuration Address Register

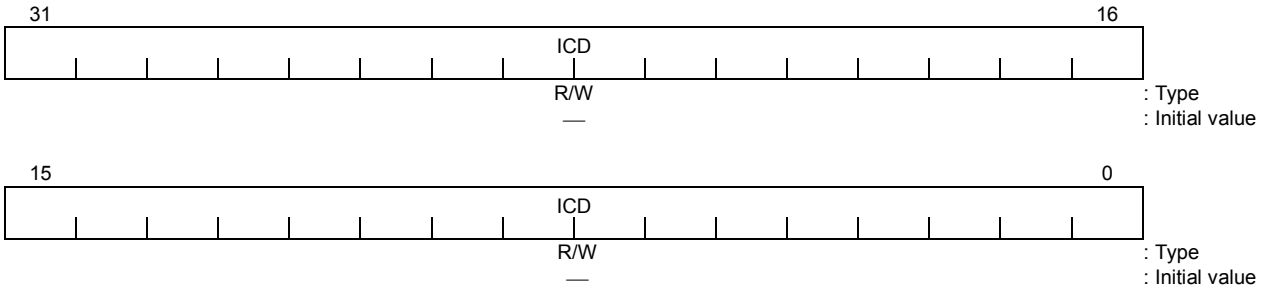
10.4.51 G2P Configuration Data Register (G2PCFGDATA) 0xD1A4

This is the only register that supports Byte access and 16-bit Word access. The upper address bit of the PCI Configuration Space is specified by the G2P Configuration Address Register (G2PCFGADRS). The lower two bits of the address are specified by the lower two bits of the offset address in this register as shown in Figure 10.4.2.

The operation of any access to this register is undefined when the PCI Controller is in the Satellite mode.

Table 10.4.2 PCI Configuration Space Access Address

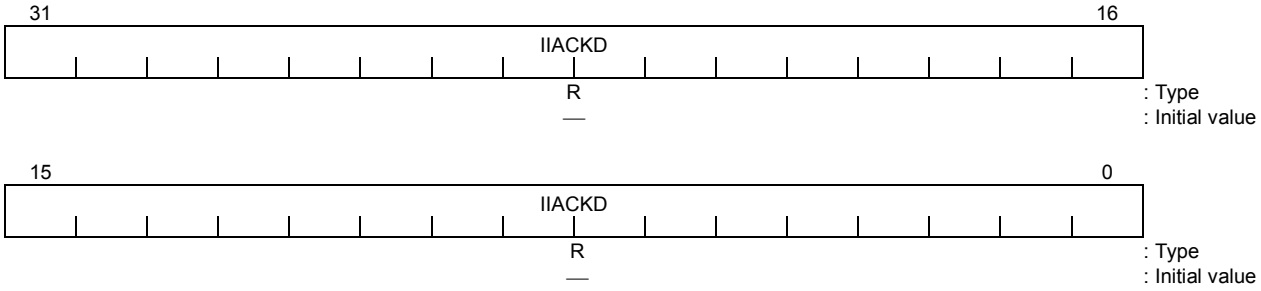
Access Size	Configuration Space Address [1:0]	Offset Address	
		Little Endian Mode	Big Endian Mode
32-bit	00	0xD1A4	0xD1A4
16-bit	00	0xD1A4	0xD1A6
	10	0xD1A6	0xD1A4
8-bit	00	0xD1A4	0xD1A7
	01	0xD1A5	0xD1A6
	10	0xD1A6	0xD1A5
	11	0xD1A7	0xD1A4



Bits	Mnemonic	Field Name	Description	Read/Write
31:0	ICD	Initiator Configuration Data	Initiator Configuration Data Register (Default--) This is a data port that is used when performing initiator PCI configuration access. PCI configuration Read or Write transactions are issued when this register is read to or written from.	R/W

Figure 10.4.49 G2P Configuration Data Register

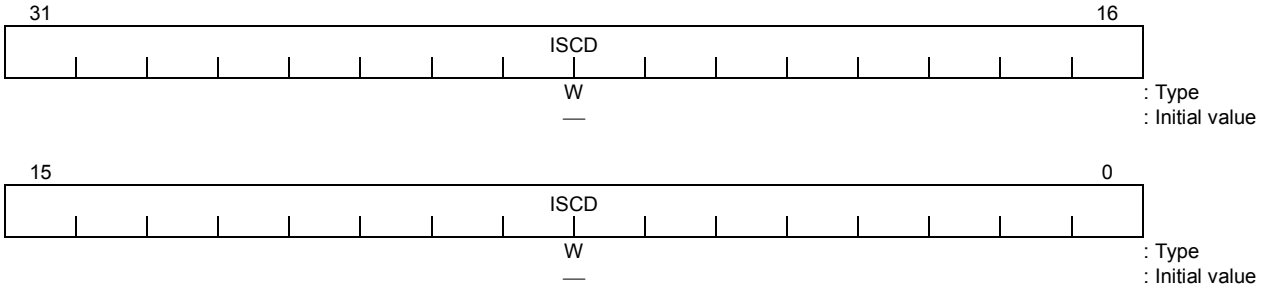
10.4.52 G2P Interrupt Acknowledge Data Register (G2PINTACK) 0xD1C8



Bits	Mnemonic	Field Name	Description	Read/Write
31:0	IIACKD	Initiator Interrupt Acknowledge Address Port	Initiator Interrupt Acknowledge Address Port (Default--) An Interrupt Acknowledge cycle is generated on the PCI Bus when this register is read. The data that is returned by this Read transaction becomes the Interrupt Acknowledge data.	R

Figure 10.4.50 G2P Interrupt Acknowledge Data Register

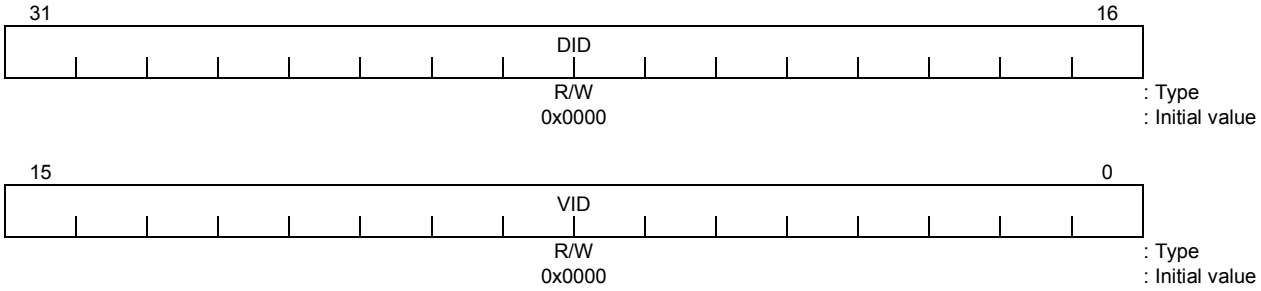
10.4.53 G2P Special Cycle Data Register (G2PSPC) 0xD1CC



Bits	Mnemonic	Field Name	Description	Read/Write
31:0	ISCD	Initiator Special Cycle Data Port	Initiator Special Cycle Data Port (Default--) When this register is written to, Special Cycles are generated on the PCI Bus depending on the data that is written.	W

Figure 10.4.51 G2P Special Cycle Data Register

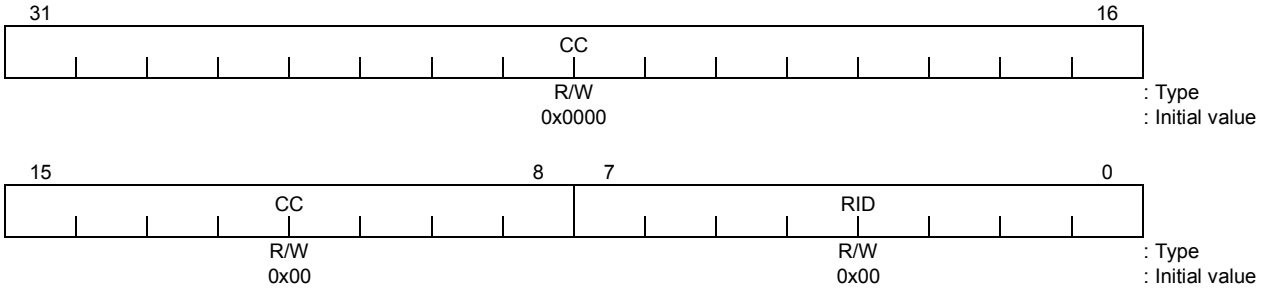
10.4.54 Configuration Data 0 Register (PCICDATA0) 0xD1D0



Bits	Mnemonic	Field Name	Description	Read/Write
31:16	DID	Device ID	Device ID (Default: 0x0000) This is the data loaded in the Device ID Register of the PCI Configuration Space.	R/W
15:0	VID	Vendor ID	Vendor ID (Default: 0x0000) This is the data loaded in the Vendor ID Register of the PCI Configuration Space.	R/W

Figure 10.4.52 ID Register

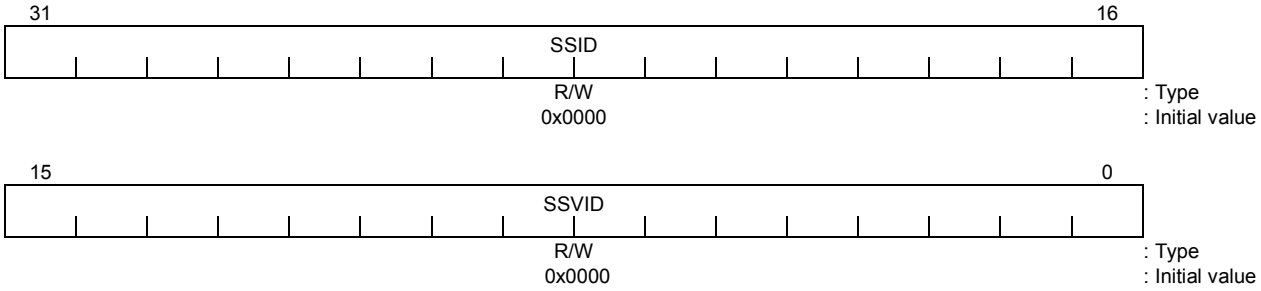
10.4.55 Configuration Data 1 Register (PCICDATA1) 0xD1D4



Bis	Mnemonic	Field Name	Description	Read/Write
31:8	CC	Class Code	Class Code (Default: 0x000000) This is the data loaded in the Class Code Register of the PCI Configuration Space.	R/W
7:0	RID	Revision ID	Revision ID (Default: 0x00) This is the data loaded in the Revision ID Register of the PCI Configuration Space.	R/W

Figure 10.4.53 Class Code/Revision ID Register

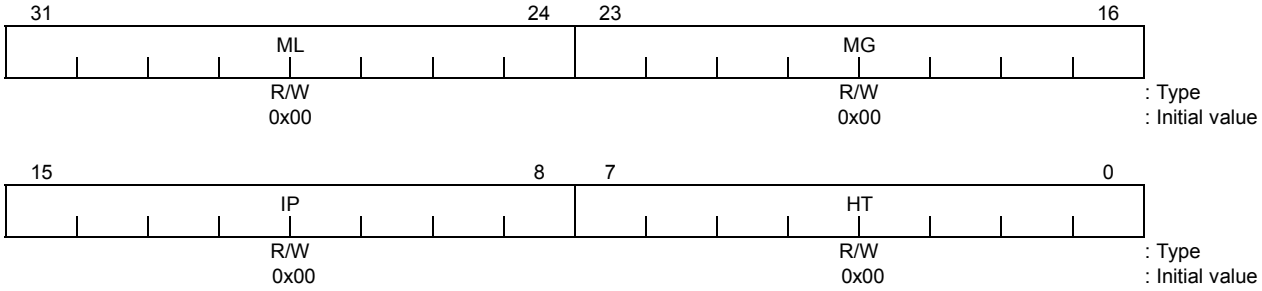
10.4.56 Configuration Data 2 Register (PCICDATA2) 0xD1D8



Bits	Mnemonic	Field Name	Description	Read/Write
31:16	SSID	Sub System ID	Subsystem ID (Default: 0x0000) This is the data loaded in the Sub System ID Register of the PCI Configuration space.	R/W
15:0	SSVID	Sub System Vendor ID	Subsystem Vendor ID (Default: 0x0000) This is the data loaded in the Sub System Vendor ID Register of the PCI Configuration space.	R/W

Figure 10.4.54 Sub System ID Register

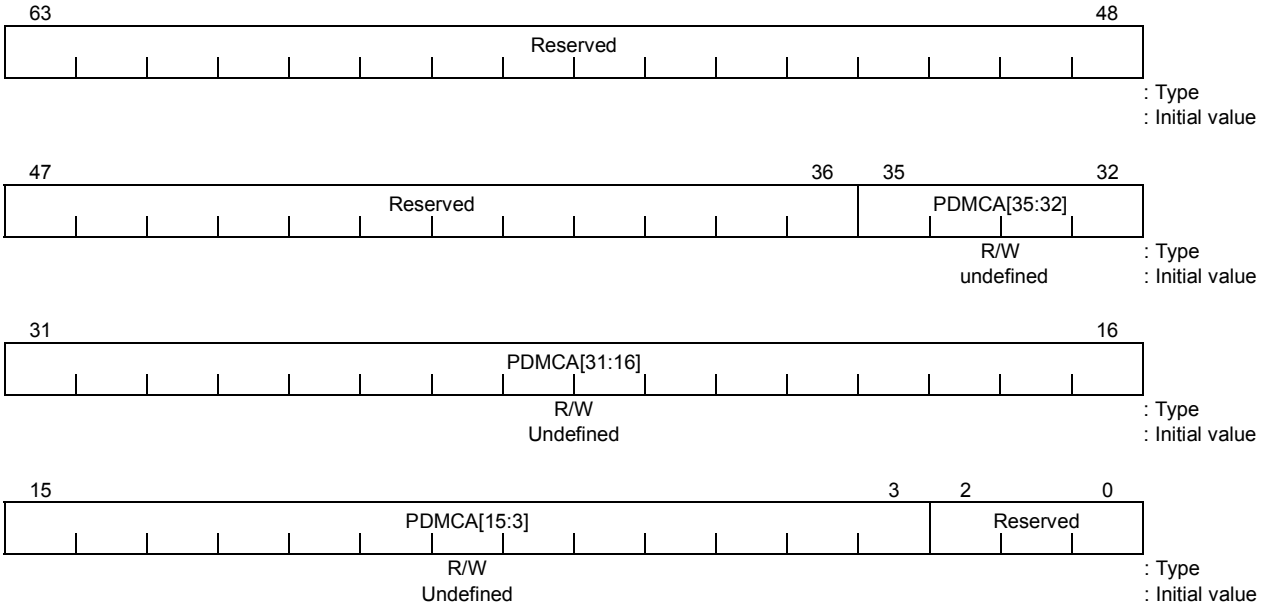
10.4.57 Configuration Data 3 Register (PCICDATA3) 0xD1DC



Bits	Mnemonic	Field Name	Description	Read/Write
31:24	ML	Maximum Latency	Max_Lat (Maximum Latency) (Default: 0x00) This is the data loaded in the Max_Lat Register of the PCI Configuration Space.	R/W
23:16	MG	Minimum Grant	Min_Gnt (Minimum Grant) (Default: 0x00) This is the data loaded in the Min_Gnt Register of the PCI Configuration Space.	R/W
15:8	IP	Interrupt Pin	Interrupt Pin (Default: 0x00) This is the data loaded in the Interrupt Pin Register of the PCI Configuration Space.	R/W
7:0	HT	Header Type	Header Type (Default: 0x00) This is the data loaded in the Header Type Register of the PCI Configuration Space.	R/W

Figure 10.4.55 PCI Configuration 2 Register

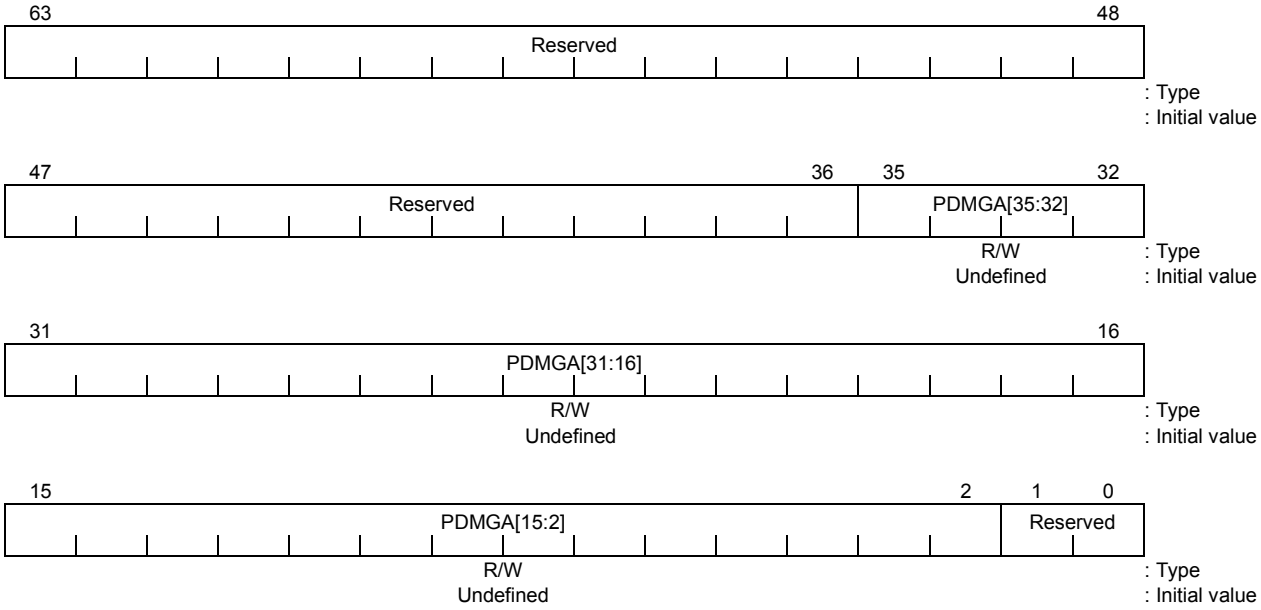
10.4.58 PDMAC Chain Address Register (PDMCA) 0xD200



Bits	Mnemonic	Field Name	Description	Read/Write
63:36		Reserved		—
35:3	PDMCA	Chain Address	PDMAC Chain Address (Default is undefined) The address of the next PDMAC Data Command Descriptor to be read is specified by a G-Bus physical address on a 64-bit address boundary. This register value is held without being affected by a Reset. 0 value judgement is performed when the lower 32 bits of this register are rewritten. DMA transfer is automatically initiated if the result is not “0”.	R/W
2:0		Reserved		—

Figure 10.4.56 PDMAC Chain Address Register

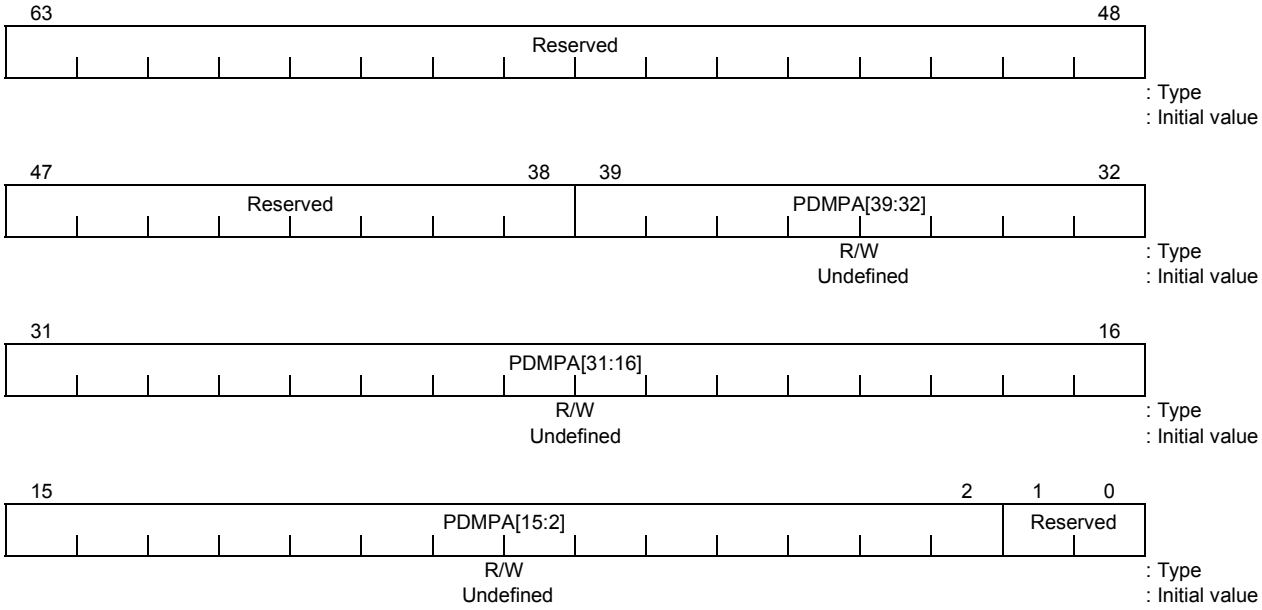
10.4.59 PDMAC G-Bus Address Register (PDMGA) 0xD208



Bits	Mnemonic	Field Name	Description	Read/Write
63:36		Reserved		—
35:2	PDMGA	G-Bus Address	PDMAC G-Bus Address (Default is undefined) The G-Bus DMA transfer address is specified by a G-Bus physical address on a 32-bit address boundary. This register value is used for G-Bus Read access during DMA transfer from the G-Bus to the PCI Bus, or it is used for G-Bus Write access during DMA transfer from the PCI Bus to the G-Bus. This register value is held without being affected by a Reset.	R/W
1:0		Reserved		—

Figure 10.4.57 G-Bus Address Register

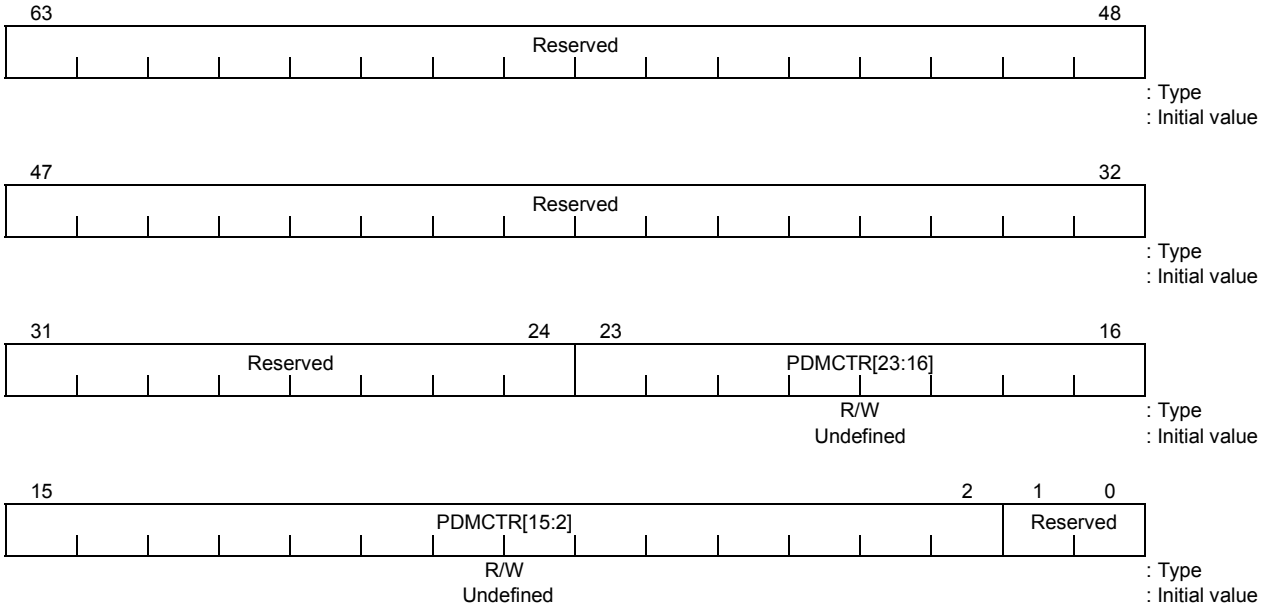
10.4.60 PDMAC PCI Bus Address Register (PDMPA) 0xD210



Bits	Mnemonic	Field Name	Description	Read/Write
63:38		Reserved		—
39:2	PDMPA	PCI Bus Address	PDMAC PCI-Bus Address (Default is undefined) The PCI Bus DMA transfer address is specified by a PCI Bus physical address on a 32-bit address boundary. This register value is held without being affected by a Reset. Note: This register value is used for PCI Bus Write access during DMA transfer from the G-Bus to the PCI Bus, or it is used for PCI Bus Read access during DMA transfer from the PCI Bus to the G-Bus.	R/W
1:0		Reserved		—

Figure 10.4.58 PCI Bus Address Register

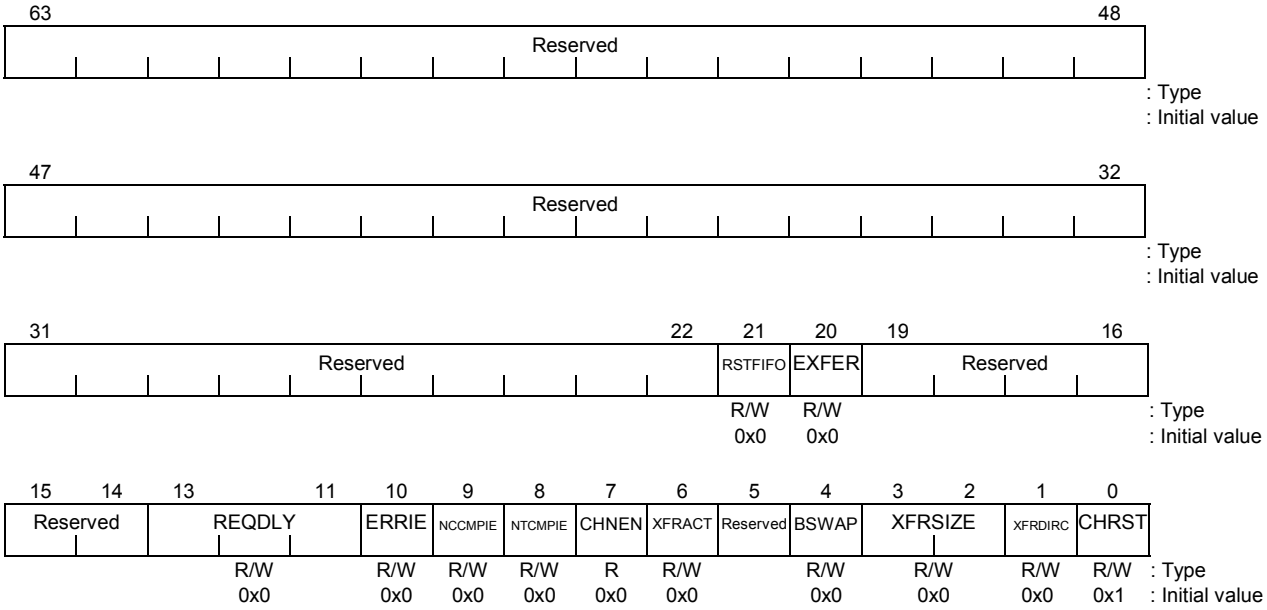
10.4.61 PDMAC Count Register (PDMCTR) 0xD218



Bits	Mnemonic	Field Name	Description	Read/Write
63:24		Reserved		—
23:2	PDMCTR	Transfer Byte Count	PDMAC Transfer Count (Default is undefined) Sets an uncoded 24-bit transfer byte count in 32-bit word units. Also, the setting of this register must always be a multiple of the transfer size specified inside the PDMAC Configuration Register. No data transfer is performed if a count of “0” is set. This byte count value is calculated from the transferred byte size as the PDMAC performs a DMA transfer. This register value is held without being affected by a Reset.	R/W
1:0		Reserved		—

Figure 10.4.59 Count Register

10.4.62 PDMAC Configuration Register (PDMCFG)0xD220



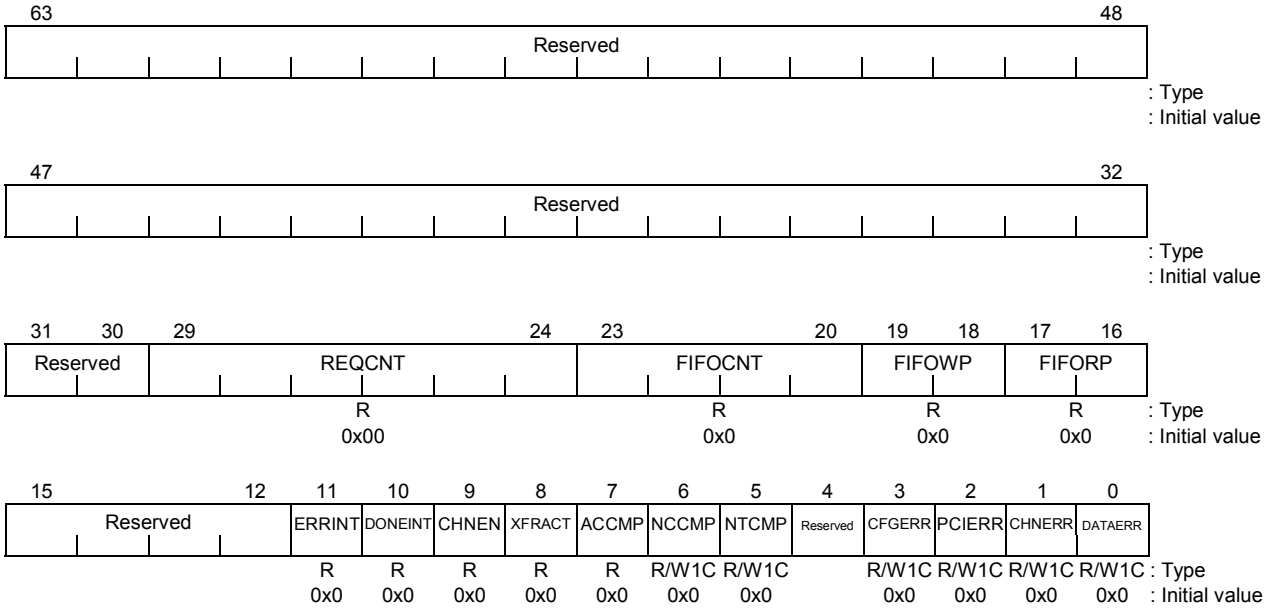
Bit	Mnemonic	Field Name	Description	Read/Write
63:22		Reserved		—
21	RSTFIFO	Reset FIFO	Reset FIFO (Default: 0x0) Initializes the Read pointer and Write pointer to the FIFO in the PDMAC, and sets the FIFO hold count to “0”. Please use the software to clear this bit when it is set. This is a function for a diagnosis. Usually, it is not used. 1: Performs FIFO reset. 0: Does not perform FIFO reset.	R/W
20	EXFER	Endian Transfer	Endian Transfer (Default: 0x0) Specifies whether to perform Endian transfer. Please use the default as is. Set up EXFER as follows according to a Endian setup of G-Bus. 1: G-Bus in Little Endian 0: G-Bus in Big Endian	R/W
19:14		Reserved		—
13:11	REQDLY	Request Delay Time	Request Delay (Default: 0x0) G-Bus transactions for DMA transfer must be performed separated at least by the interval this field specifies. 000: Continuously try to perform G-Bus transfer. 001: 16 G-Bus clocks 010: 32 G-Bus clocks 011: 64 G-Bus clocks 100: 128 G-Bus clocks 101: 256 G-Bus clocks 110: 512 G-Bus clocks 111: 1024 G-Bus clocks	R/W
10	ERRIE	Error Detect Interrupt Enable	Interrupt Enable on Error (Default: 0x0) 1: PDMAC generates an error during error detection. 0: PDMAC does not generate an error during error detection.	R/W
9	NCCMPIE	Normal Chain Complete Interrupt Enable	Interrupt Enable on Chain Done (Default: 0x0) 1: PDMAC generates an interrupt when the current chain is complete. 0: PDMAC does not generate an interrupt when the current chain is complete.	R/W

Figure 10.4.60 PDMAC Configuration Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
8	NTCMPIE	Normal Data Transfer Complete Interrupt Enable	Interrupt Enable on Transfer Done (Default: 0x0) 1: PDMAC generates an interrupt when the current data transfer is complete. 0: PDMAC does not generate an interrupt when the current data transfer is complete.	R/W
7	CHNEN	Chain Enable	Chain Enable (Default: 0x0) (Read Only) When the current data transfer is complete, this field reads the next data command Descriptor from the address indicated by the PDMAC Chain Address Register then indicates whether to continue the transfer or not. This bit is only set to “1” when either a CPU Write process or a Descriptor Read process sets a value other than “0” in the PDMAC Chain Address Register. This bit is cleared to “0” if either the Channel Reset bit is set, or “0” is set in the PDMAC Chain Address Register by a CPU Write or a Descriptor Read process. The above 0 value judgement is not performed when the TX49/H3 core stores the upper 32 bits in the PDMAC Chain Address Register. 1: Reads the next data command Descriptor. 0: Does not read the next data command Descriptor.	R
6	XFRACT	Transfer Active	Transfer Active (Default: 0x0) Specifies whether to perform DMA transfer or not. Setting this bit after setting the appropriate value in the register group initiates DMA data transfer. This bit is not set if the PDMAC Count Register value is “0” and the Chain Enable bit is cleared when “1” is written to this bit. Even when a value other than “0” is written to the Chain Address Register, “1” is set to this bit and DMA transfer automatically starts. The above 0 value judgement is not performed when the TX49/H3 core stores the upper 32 bits in the PDMAC Chain Address Register. Data transfer will be stopped after a short delay if this bit is cleared while the data transfer is in progress. This bit is automatically cleared to “0” either when data transfer ends normally or is stopped by an error. Never clear XFRACT by software, because it stops guaranteeing a normal operation,. 1: Perform data transfer. 0: Do not perform data transfer.	R/W
5		Reserved		—
4	BSWAP	Byte Swap Within DWORD	Swap Bytes in DWORD (Default: 0x0) Specifies whether to perform 32-bit data byte swapping. Please leave this bit at “0” for normal usage. Setting this bit when in the Big Endian mode executes data transfer so the byte order of the 32-bit data on the PCI Bus (which is Little Endian) does not change. 1: Swap the byte order of each 32-bit DWORD data, then transfer. 0: Transfer without swapping the byte order of each 32-bit DWORD data.	R/W
3:2	XFRSIZE	Transfer Size	Transfer Size (Default: 0x0) Specifies the data transfer size in one G-Bus transaction on the G-Bus. 00: 1 DWORD (32-bit) 01: 1 QWORD (64-bit) 10: 4 QWORD (Burst transfer) 11: Reserved	R/W
1	XFRDIRC	Transfer Direction	Transfer Direction (Default: 0x0) Specifies the DMA data transfer direction. 1: Transfers data from the G-Bus to the PCI Bus. 0: Transfers data from the PCI Bus to the G-Bus.	R/W
0	CHRST	Channel Reset	Channel Reset (Default: 0x1) Resets the DMA channel. This bit must be cleared by the software in advance so the channel can start the data transfer. This reset function is not supported when PDMAC is in operation. Ensure that the Transfer Active (XFARCT) bit in the PDMSTATUS register is cleared prior to resetting the DMA channel. For chained DMA, also ensure either the Abnormal Chain Complete (ACCMP) or Normal Chain Complete (NCCMP) bit in the PDMSTATUS register is set. 1: All logic and State Machines are reset. 0: The channel becomes valid.	R/W

Figure 10.4.60 PDMAC Configuration Register (2/2)

10.4.63 PDMAC Status Register (PDMSTATUS) 0xD228



Bit	Mnemonic	Field Name	Description	Read/Write
63:30		Reserved		—
29:24	REQCNT	Request Delay Time Counter	Request Delay Counter (Default: 0x00) This field indicates the request delay time counter value as 16 × *n* when the 6-bit value of this field is *n*.	R
23:20	FIFOCNT	FIFO Hold Count	FIFO Valid Entry Count (Default: 0x0) This field indicates the number of bytes that was written in the FIFO but not yet read. This is a diagnostic function.	R
19:18	FIFOWP	FIFO Write Pointer	FIFO Write Pointer (Default: 0x0) This field indicates the next Write position in the FIFO. This is a diagnostic function.	R
17:16	FIFORP	FIFO Read Pointer	FIFO Read Pointer (Default: 0x0) This field indicates the next Read position in the FIFO. This is a dianostic function.	R
15:12		Reserved		—
11	ERRINT	Error Interrupt Status	Error Interrupt Status (Default: 0x0) Indicates whether to signal an error interrupt. 1: An error interrupt request exists. 0: No error interrupt request exists.	R
10	DONEINT	Normal Transfer Complete Interrupt Status	Normal Transfer Complete Interrupt Status (Default: 0x0) Indicates whether a Normal Transfer Complete Interrupt is signaled. This bit becomes “1” when either the Normal Chain Complete bit (NCCMP) is set and the Normal Chain Complete Interrupt Enable bit (NCCMPIE) is set, or when the Normal Data Transfer Complete bit (NTCMP) is set and the Normal Data Transfer Complete Interrupt Enable bit (NTCMPIE) is set. 1: A Normal Transfer Complete Interrupt request exists. 0: No Normal Transfer Complete Interrupt request exists.	R
9	CHNEN	Chain Enable	Chain Enable (Default: 0x0) This bit is a copy of the Chain Enable bit in the PDMAC Configuration Register.	R

Figure 10.4.61 Status Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
8	XFRACT	Transfer Active	Transfer Active (Default: 0x0) This bit is a copy of the Transfer Active bit in the PDMAC Configuration Register.	R
7	ACCMP	Abnormal Chain Completion	Abnormal Chain Complete (Default: 0x0) 1: Indicates that the Chain transfer ended in an error state. In other words, this reflects an OR operation of the PDMAC Status Register bits [3:0]. 0: Indicates that no error has occurred in the Chain transfer since the previous error bit was cleared. Note: Bits [3:0] of the PDMAC Status Register must be cleared in order to clear this bit.	R
6	NCCMP	Normal Chain Completion	Normal Chain Complete (Default: 0x0) 1: Indicates that the Chain transfer ended in the Normal state. 0: Indicates that Chain transfer has not ended since this bit was previously cleared.	R/W1C
5	NTCMP	Normal Data Transfer Complete	Normal Data Transfer Complete (Default: 0x0) 1: Indicates that the data transfer specified by the PDMAC Register ended in the Normal state. 0: Indicates that data transfer has not ended since this bit was previously cleared.	R/W1C
4		Reserved		—
3	CFGERR	Configuration Error	Configuration Error (Default: 0x0) 1: Indicates that either the current setting of the control portion in the Control Register and the Address/Count Register are not consistent with each other or the PDMAC stipulation is not being obeyed. DMA transfer stops. 0: Indicates that the current setting of the control portion in the Control Register can be tolerated.	R/W1C
2	PCIERR	PCI Fatal Error	PCI Fatal Error (Default: 0x0) 1: Indicates that an error was signaled on the PCI Bus during the Chain process. 0: Indicates that no error has been signaled on the PCI Bus since this bit was previously cleared.	R/W1C
1	CHNERR	G-Bus Chain Error	G-Bus Chain Bus Error (Default: 0x0) 1: Indicates that a G-Bus error occurrred during the Chain process. DMA transfer stops. 0: Indicates that no G-Bus error has occurred during the Chain process since this bit was cleared.	R/W1C
0	DATAERR	G-Bus Data Error	G-Bus Data Bus Error (Default: 0x0) 1: Indicates that a G-Bus error occurred during the data transfer process. DMA transfer stops. 0: Indicates that no G-Bus error has occurred during the data transfer process since this bit was cleared.	R/W1C

Figure 10.4.61 Status Register (2/2)

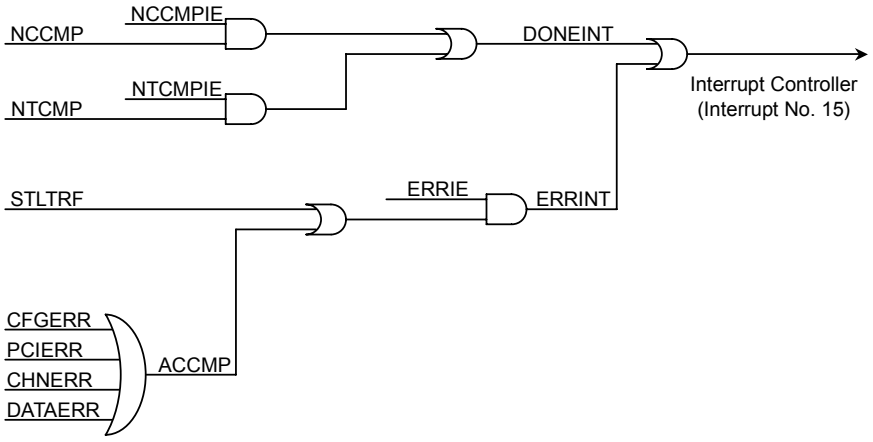


Figure 10.4.62 PDMAC Interrupt Signaling

10.5 PCI Configuration Space Register

The PCI Configuration Space Register is accessed using PCI Configuration cycles by way of an external PCI host device only when in the Satellite mode. Table 10.5.1 lists registers contained within the PCI Configuration Space Register. The registers in the table with a shaded background are those whose values can be rewritten using EEPROM. (See 10.3.14.)

Registers at addresses 0x00 through 0x41 can use the corresponding PCI Controller Control Register to access from the TX49/H3 core when in the Host mode. Please refer to the explanation of the corresponding PCI Controller Control registers for more information about these registers. This section only describes the registers that are accessed from the PCI Configuration Space.

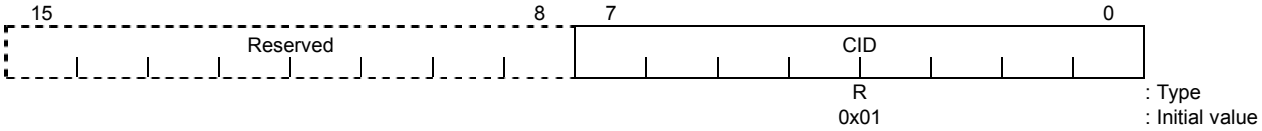
Also, it is possible to read some of the fields in the Status Register and PMCSR register from the Satellite Mode PCI Status Register.

Please refer to the PCI Bus Specifications for more information on the PCI Configuration Register.

Table 10.5.1 PCI Configuration Space Register

Address	31 16	15 0	Corresponding Register
00h	Device ID	Vendor ID	PCIID
04h	Status	Command	PCISTATUS
08h	Class Code	Revision ID	PCICCREV
0Ch	BIST / Header Type	Latency Timer / Cache Line Size	PCICFG1
10h	Memory Space 0 Lower Base Address		P2GM0PLBASE
14h	Memory Space 0 Upper Base Address		P2GM0PUBASE
18h	Memory Space 1 Lower Base Address		P2GM1PLBASE
1Ch	Memory Space 1 Upper Base Address		P2GM1PUBASE
20h	Memory Space 2 Base Address		P2GM2PBASE
24h	I/O Space Base Address		P2GIOPBASE
28h	Reserved		—
2Ch	Subsystem ID	Subsystem Vendor ID	PCISID
30h	Reserved		—
34h	Reserved	Capabilities Pointer (Cap_Ptr)	PCICAPPTR
38h	Reserved		—
3Ch	Max_Lat / Min_Gnt	Interrupt Pin / Interrupt Line	PCICFG2
40h	Reserved	Retry Timeout Value / TRDY Timeout Value	G2PTOCNT
44h-DBh	Reserved		—
DCh	Power Management Capabilities (PMC)	Next Item Ptr (Next_Item_Ptr) / Capability ID (Cap_ID)	—
E0h	Reserved / Reserved	Power Management Control/Status Register (PMCSR)	—
E4h-FFh	Reserved		—

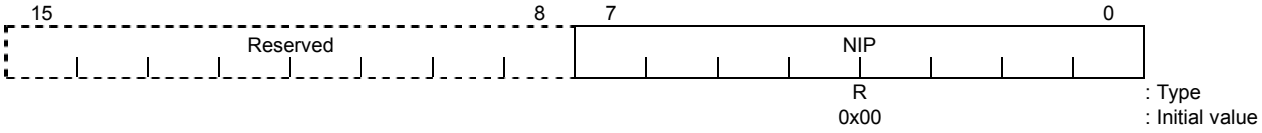
10.5.1 Capability ID Register (Cap_ID) 0xDC



Bits	Mnemonic	Field Name	Description	Read/Write
15:8		Reserved		—
7:0	CID	Capability ID	Capability ID (Default: 0x01) Indicates that a list is the link list of the Power Management Register.	R

Figure 10.5.1 Capability ID Register

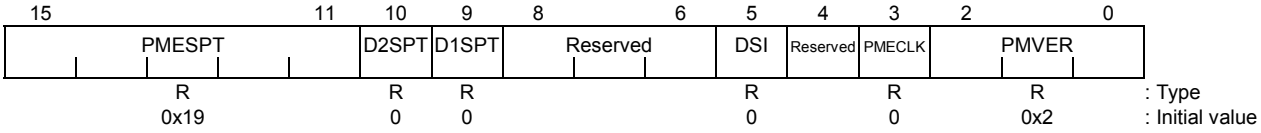
10.5.2 Next Item Pointer Register (Next_Item_Ptr) 0xDD



Bits	Mnemonic	Field Name	Description	Read/Write
15:8		Reserved		—
7:0	NIP	Next Item Pointer	Next Item Pointer (Default: 0x0) This is the Next Item pointer. Indicates the end of a list.	R

Figure 10.5.2 Next Item Pointer Register

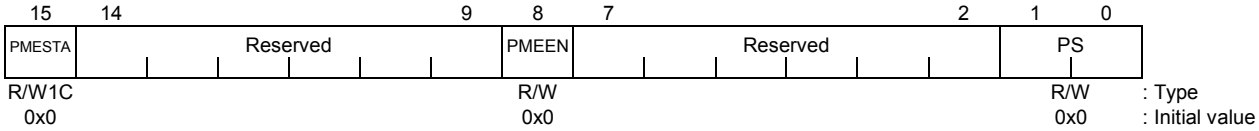
10.5.3 Power Management Capability Register (PMC) 0xDE



Bit	Mnemonic	Field Name	Description	Read/Write
15:11	PMESPT	PME Output Support	PME_ Support (Fixed Value: 0x09) Indicates that the PME* signal can be output from the state where the bit is set to “1”. Bit 15: Can output the PME* signal from the D3cold state. Bit 14: Can output the PME* signal from the D3hot state. Bit 13: Can output the PME* signal from the D2 state. Bit 12: Can output the PME* signal from the D1 state. Bit 11: Can output the PME* signal from the D0 state. Note: With the TX4938 PCI Controller, it is possible to output the PME* signal from the D0 and the D3hot states.	R
10	D2SPT	D2 Support	D2_Support (Fixed Value: 0) 0: Indicates that the D2 state is not supported.	R
9	D1SPT	D1 Support	D1_Support (Fixed Value: 0) 0: Indicates that the D1 state is not supported.	R
8:6		Reserved		—
5	DSI	DSI	DSI (Fixed Value: 0) 0: Indicates that Device Specific Initialization is not required.	R
4		Reserved		—
3	PMECLK	PME Clock	PME Clock (Fixed Value: 0) 0: Indicates that the PCI Clock is not required to assert the PME* signal.	R
2:0	PMVER	Power Management I/F Version	Version (Fixed Value: 0x2) 2: Indicates compliance with "PCI Power Management Interface Specification" Version 1.1.	R

Figure 10.5.3 PMC Register

10.5.4 Power Management Control/Status Register (PMCSR) 0xE0



Bit	Mnemonic	Field Name	Description	Read/Write
15	PMESTA	PME Status	PME_Status (Default: 0x0) Indicates the existence of a PME (Power Management Event) . 1: There is a PME. 0: There is no PME. The value of this bit becomes “1” when Writing a “1” to the PME bit (P2GCFG.PME) of the P2G Configuration Register. This bit is cleared when the Host Bridge writes a “1”. It is possible to signal a PME* Clear Interrupt at this time.	R/W1C
14:9		Reserved		—
8	PMEEN	PME Enable	PME_En (Default: 0x0) Sets PME* signal assertion to enable or disable. 1: Enables assertion of the PME* signal. 0: Disables assertion of the PME* signal. The PME_En set bit of the P2G Status Register (P2GSTATUS.PMEES) is set when this bit is set. At this time, it is possible to signal the PME_En set interrupt.	R/W
7:2		Reserved		—
1:0	PS	Power State	PowerState (Default: 0x0) Sets the Power Management state. The Power Management State Change bit (P2GSTATUS.PMSC) of the P2G Status Register is set when the value of this field is changed. It also becomes possible to generate a Power State Change Interrupt at this time. The TX4938 can read the value of this field from the PowerState field (PCISSTATUS.PS) of the Satellite Mode PCI Status Register. 00b: D0 (no change) 01b: D1 :Reserved 10b: D2 :Reserved 11b: D3hot	R/W

Figure 10.5.4 PMCSR Register

11. Serial I/O Port

11.1 Features

The TX4938 asynchronous Serial I/O (SIO) interface has two full duplex UART channels (SIO0 and SIO1). SIO has the following features.

(1) Full duplex transmission (simultaneous transmission and reception)

(2) On-chip baud rate generator

(3) Modem flow control (CTS/RTS)

(4) FIFO

- Transmit FIFO: 8 bits × 8 stages
- Reception FIFO: 13 bits × 16 stages (data: 8 bits, status: 5 bits)

(5) Supports DMA transfer

(6) Supports multi-controller systems

- Supports Master/Slave operation

11.2 Block Diagram

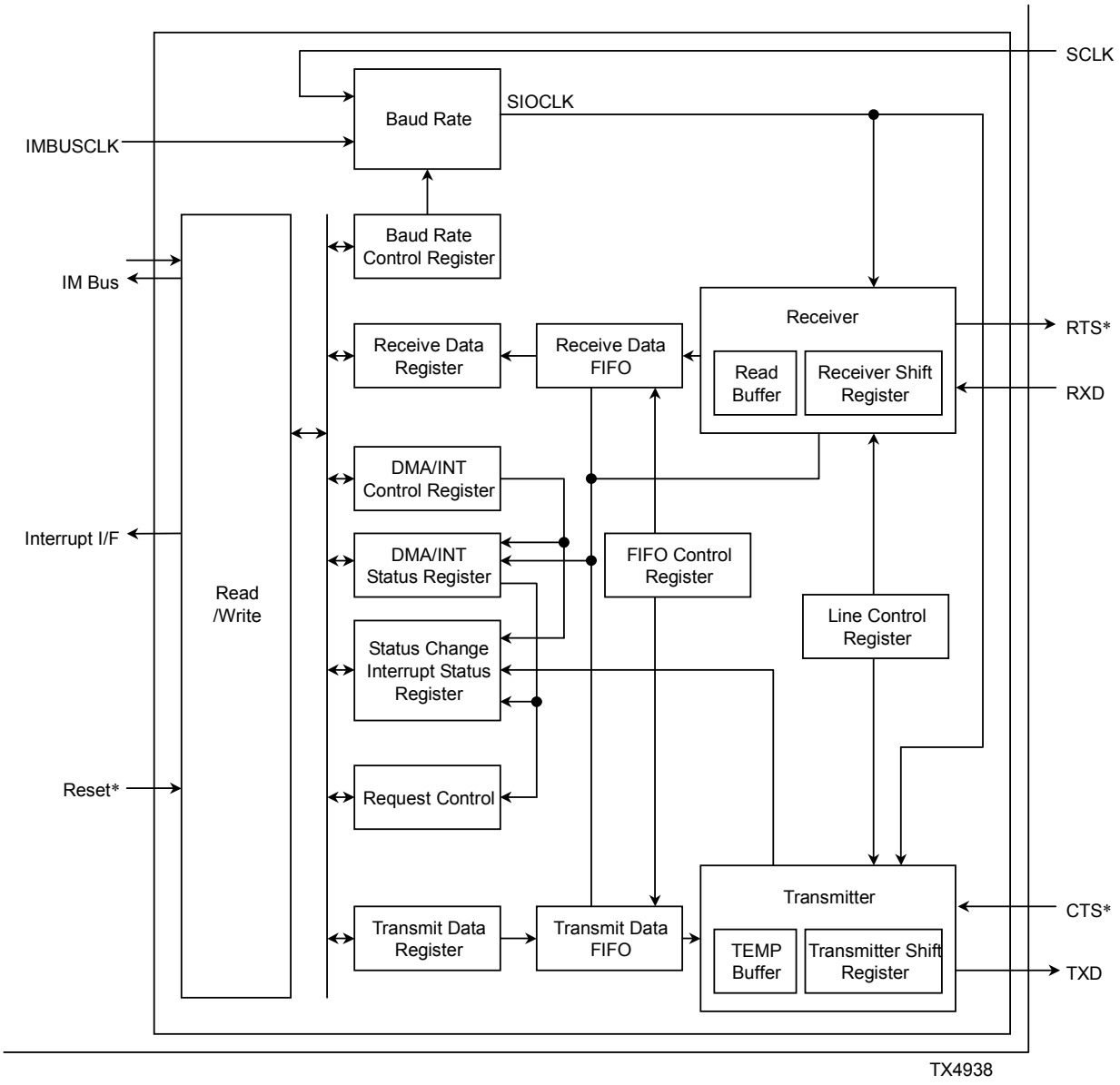


Figure 11.2.1 SIO Internal Block Diagram

11.3 Detailed Explanation

11.3.1 Overview

During reception, serial data that are input as an RXD signal from an external source are converted into parallel data, then are stored in the Receive FIFO buffer. Parallel data stored in the FIFO buffer are fetched by either CPU or DMA transfer.

During transmission, parallel data written to the Transmit FIFO buffer by CPU or DMA transfer are converted into serial data, then are output as a TXD signal.

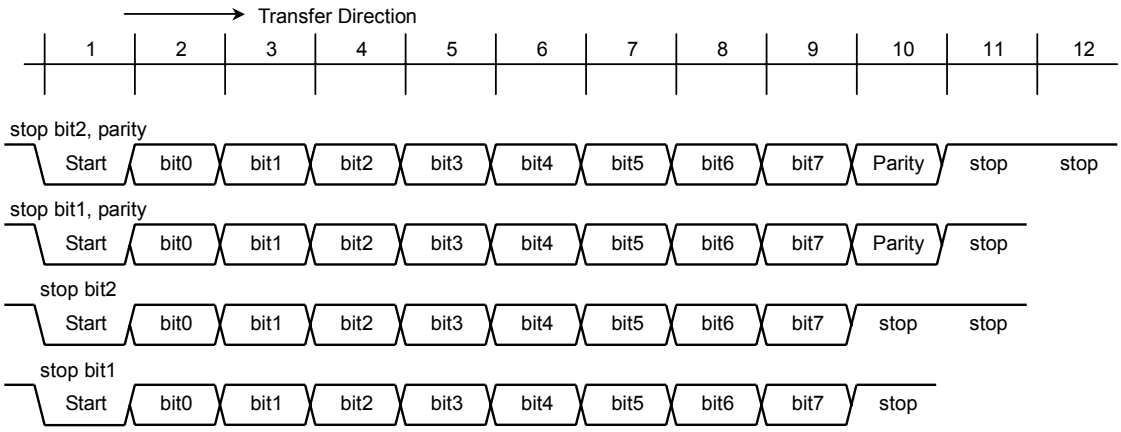
11.3.2 Data Format

The TX4938 SIO can use the following data formats.

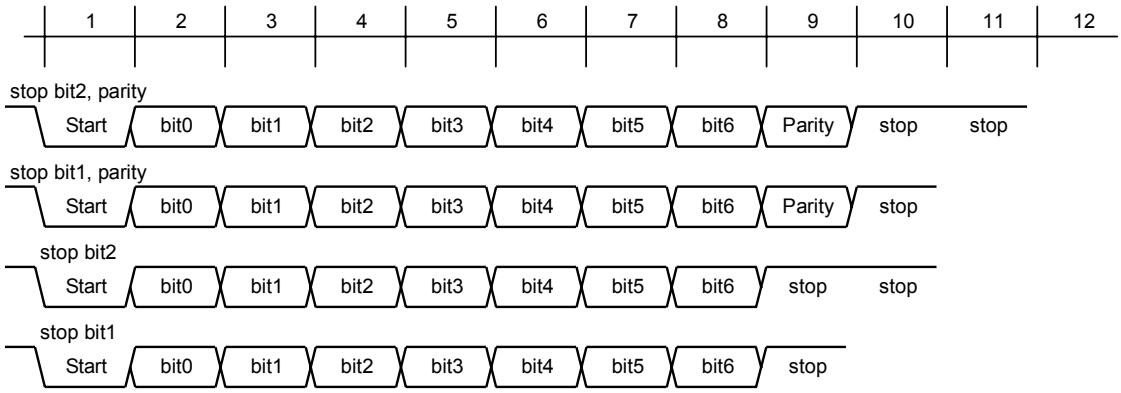
Data Length	: 8/7 bits
Stop Bit	: 1/2 bits
Parity Bit	: Yes/No
Parity Format	: Even/Odd
Start Bit	: Fixed to 1 bit

Figure 11.3.1 illustrates the data frame when making each setting.

8-bit Data

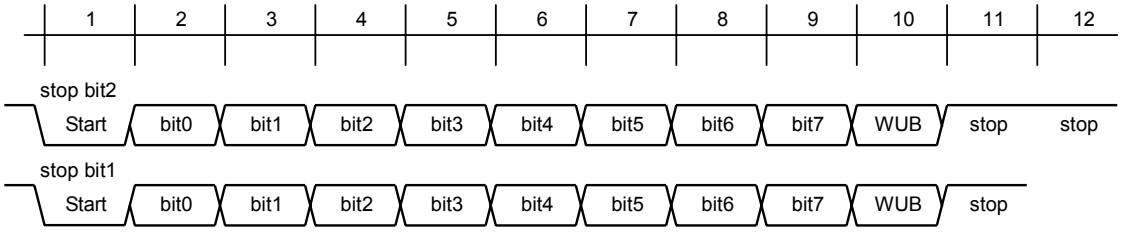

Transfer Direction
1 2 3 4 5 6 7 8 9 10 11 12
stop bit2, parity
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 Parity stop stop
stop bit1, parity
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 Parity stop
stop bit2
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 stop stop
stop bit1
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 stop

7-bit Data


1 2 3 4 5 6 7 8 9 10 11 12
stop bit2, parity
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 Parity stop stop
stop bit1, parity
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 Parity stop
stop bit2
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 stop stop
stop bit1
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 stop

8-bit Data Multi-Control System

WUB = Wake Up bit
1: Address (ID) Frame
0: Data Frame


1 2 3 4 5 6 7 8 9 10 11 12
stop bit2
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 WUB stop stop
stop bit1
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 WUB stop

7-bit Data Multi-Control System

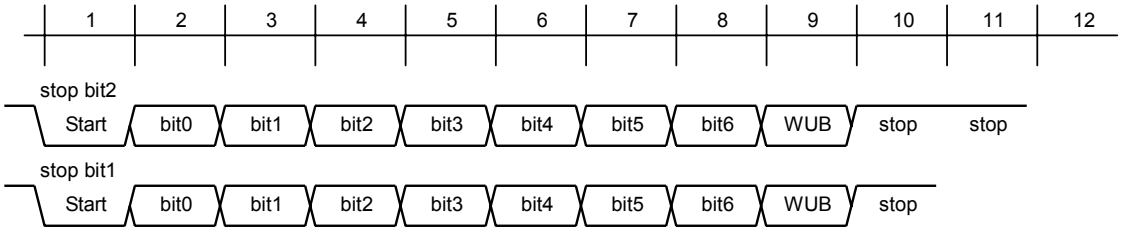

1 2 3 4 5 6 7 8 9 10 11 12
stop bit2
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 WUB stop stop
stop bit1
Start bit0 bit1 bit2 bit3 bit4 bit5 bit6 WUB stop

Figure 11.3.1 Data Frame Configuration

11.3.3 Serial Clock Generator

Generates the Serial Clock (SIOCLK). SIOCLK determines the serial transfer rate and has a frequency that is 16× the baud rate. One of the following can be selected as the source for the Serial Clock (SIOCLK).

- Internal System Clock (IMBUSCLK)
- External Clock Input (SCLK)
- Baud rate generator circuit output

The IMBUSCLK frequency can be selected from frequencies that are 1/2, 1/2.5, 1/3, or 1/4 the frequency of the CPU clock. The maximum frequency tolerance of the external clock input (SCLK) is 45% the frequency of IMBUSCLK. For example, if IMBUSCLK = 60 MHz, then set SCLK to 27 MHz or less.

The baud rate generator is a circuit that divides these clock signals according to the following formula.

$$\text{Baud Rate} = \frac{\text{fc}}{\text{Prescalar} \times \text{Divisor} \times 16}$$

- fc: Clock frequency of IMBUSCLK or an external clock input (SCLK)
- Prescalar Value: 2, 8, 32, 128
- Divide Value: 1, 2, 3,…255

Table 11.3.1 shows example settings of divide values relative to representative baud rates.

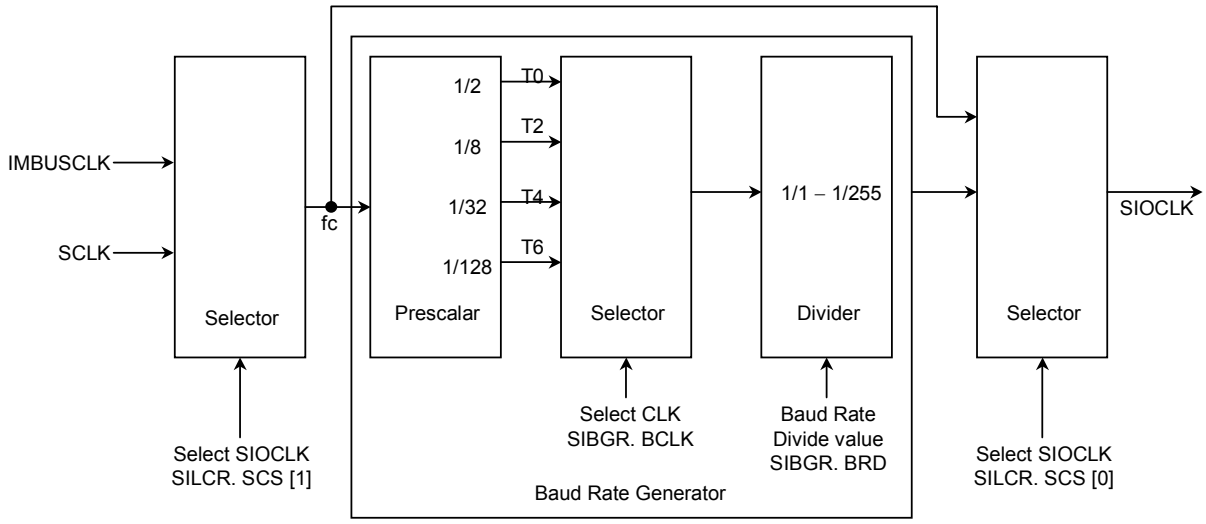


Figure 11.3.2 Baud Rate Generator and SIOCLK Generator

It is possible to correctly receive data if the error of the baud rate set by this controller is within 3.12 % of the target baud rate (communication baud rate).

Table 11.3.1 Example Divide Value Settings (and error [%] from target baud rate value)

fc[MHz]		kbps	Prescalar Value (SIBGR.BLCK) and Divide Value (SIBGR.BRD)							
			2		8		32		128	
IMBUSCLK	66	0.11								
		0.15							215	-0.07%
		0.30							107	0.39%
		0.60					215	-0.07%	54	-0.54%
		1.20					107	0.39%	27	-0.54%
		2.40			215	-0.07%	54	-0.54%	13	3.29%
		4.80			107	0.39%	27	-0.54%	7	-4.09%
		9.60	215	-0.07%	54	-0.54%	13	3.29%		
		14.40	143	0.16%	36	-0.54%	9	-0.54%		
		19.20	107	0.39%	27	-0.54%	7	-4.09%		
		28.80	72	-0.54%	18	-0.54%				
		38.40	54	-0.54%	13	3.29%				
		57.60	36	-0.54%	9	-0.54%				
		76.80	27	-0.54%	7	-4.09%				
		115.20	18	-0.54%						
	60	0.11							255	4.45%
		0.15							195	0.16%
		0.30							98	-0.35%
		0.60					195	0.16%	49	-0.35%
		1.20					98	-0.35%	24	1.73%
		2.40			195	0.16%	49	-0.35%	12	1.73%
		4.80			98	-0.35%	24	1.73%	6	1.73%
		9.60	195	0.16%	49	-0.35%	12	1.73%		
		14.40	130	0.16%	33	-1.36%	8	1.73%		
		19.20	98	-0.35%	24	1.73%	6	1.73%		
		28.80	65	0.16%	16	1.73%				
		38.40	49	-0.35%	12	1.73%				
		57.60	33	-1.36%	8	1.73%				
		76.80	24	1.73%	6	1.73%				
		115.20	16	1.73%						
SCLK	7.373	0.11					131	-0.07%	33	-0.82%
		0.15					96	0.00%	24	0.00%
		0.30					48	0.00%	12	0.00%
		0.60			96	0.00%	24	0.00%	6	0.00%
		1.20			48	0.00%	12	0.00%	3	0.00%
		2.40	96	0.00%	24	0.00%	6	0.00%		
		4.80	48	0.00%	12	0.00%	3	0.00%		
		9.60	24	0.00%	6	0.00%				
		14.40	16	0.00%	4	0.00%	1	0.00%		
		19.20	12	0.00%	3	0.00%				
		28.80	8	0.00%	2	0.00%				
		38.40	6	0.00%						
		57.60	4	0.00%	1	0.00%				
		76.80	3	0.00%						
		115.20	2	0.00%						

11.3.4 Data Reception

When the Serial Data Reception Disable bit (RSDE) of the Flow Control Register (SIFLCRn) is set to "0", reception operation starts after the RXD signal start bit is detected. Start bits are detected when the RXD signal transitions from the High state to the Low state. Therefore, the RXD signal is not interpreted as a start bit if it is Low when the Serial Data Reception Disable bit is set to "0".

The received data are stored in the Receive FIFO. The Reception Data Full bit (RDIS) of the DMA/Interrupt Status Register (SIDISRn) is set if the byte count of the stored reception data exceeds the value set by the Receive FIFO Request Trigger Level field (RDIL) of the FIFO Control Register (SIFCRn).

An interrupt is signaled when the Reception Data Interrupt Enable bit (RIE) of the DMA/Interrupt Control Register (SIDICRn) is set. The received data can be read from the Receive FIFO Data Register (SIRFIFOn).

In addition, DMA transfer is initiated when the Reception Data DMA Enable bit (RDE) of the DMA/Interrupt Control Register (SIDICRn) is set.

11.3.5 Data Transmission

Data stored in the Transmission Data FIFO are transmitted when the Serial Data Transmission Disable bit (TSDE) of the Flow Control Register (SIFLCRn) is set to "0".

If the available space in the Transmit FIFO is equal to or greater than the byte count set by the Transmit FIFO Request Trigger Level (TDIL) of the Control Register (SIFCRn), the transmission data empty bit (TDIS) of the DMA/Interrupt Status Register (SIDISRn) is set.

An interrupt is signaled when the Transmission Data Interrupt Enable bit (TIE) of the DMA/Interrupt Control Register (SIDICRn) is set.

In addition, DMA transfer is initiated when the Transmission Data DMA Enable bit (TDE) of the DMA/Interrupt Control Register (SIDICRn) is set.

11.3.6 DMA Transfer

The DMA Request Select field (INTDMA[7:0]) of the Pin Configuration Register (PCFG) can be used to allocate DMA channels for each reception and transmission channel in the following manner.

SIO Channel 1 Reception	DMA Channel 0
SIO Channel 1 Transmission	DMA Channel 1
SIO Channel 0 Reception	DMA Channel 2
SIO Channel 0 Transmission	DMA Channel 3

Set the DMA Channel Control Register of the DMA Controller as described below.

DMA Request Polarity	Low Active	DMCCRn.ACKPOL = 0
DMA Acknowledge Polarity	Low Active	DMCCRn.REQPOL = 0
Request Detection	Level Detection	DMCCRn.EGREQ = 0

Transfer Size	1 Byte	DMCCRn.XFSZ = 000b
Transfer Address Mode	Dual	DMCCRn.SNGAD = 0

In the case of transmission channels, the address of the Transmit FIFO Register (SITFIFOn) is set in the DMAC Destination Address Register (DMDARn). In the case of reception channels, the address of the Receive FIFO Register (SIRFIFOn) is set in the DMAC Source Address Register (DMSARn). Please set the addresses specified in "11.4.8 Transmit FIFO Register" and "11.4.9 Receive FIFO Register" since the set address differs depending on the Endian mode.

11.3.7 Flow Control

SIO supports hardware flow control that uses the RTS*/CTS* signal.

The CTS* (Clear to Send) input signal indicates that data can be received from the reception side when it is Low. Setting the Transmission Enable Select bit (TES) of the Flow Control Register (SIFLCRn) makes transmission flow control that uses the CTS* signal more effective.

It is also possible to generate status change interrupts by changing the state of the CTS* signal. The conditions in which interrupts are generated can be selected by the CTSS Active Condition field of the DMA/Interrupt Control Register (SIDICRn).

Setting the RTS* (Request to Send) output signal to High requests the transmission side to pause transmission. Transmission resumes when the reception side becomes ready and the RTS* signal is set to Low.

Setting the Reception Enable Select bit (RCS) of the flow Control Register (SIFLCRn) makes reception flow control that uses the RTS* signal more effective. The RTS* signal pin status becomes High when data of the byte count set by the RTS Active Trigger Level field (RTSTL) of the Flow Control Register (SIFLCRn) accumulates in the Receive FIFO. The RTS* signal can also be made High by setting the RTS Software Control bit (RTSSC) of the Flow Control Register (SIFLCRn). Setting this bit requests the transmission side to pause transmission.

11.3.8 Reception Data Status

Status data such as the following is also stored in the Receive FIFO.

- Overrun error

  An overrun error is generated if all 16-stage Receive FIFO buffers become full and more data is transferred to the Reception Read buffer. When this occurs, the Overrun Status bit is set by the last stage of the Receive FIFO.

- Parity error

  A parity error is generated when a parity error is detected in the reception data.

- Framing error

  A framing error is generated when "0" is detected at the first stop bit of the reception data.

- Break reception

  A break is detected when a framing error occurs in the reception data and all data in a single frame are "0". When this occurs, 2 frames (2 Bytes) of 0x00 data are stored in the Receive FIFO.

The Reception Error Interrupt bit (SIDISR.ERI) of the DMA/Interrupt Status Register (SIDISRn) is set when one of the following errors is detected: an overrun error, a parity error, or a framing error. An interrupt is signaled if the Reception Error Interrupt Enable bit of the DMA/Interrupt Control Register (SIDICRn) is set.

The Break Detected bit (UBRKD) and the Receiving Break bit (RBRKD) of the Status Change Interrupt Status Register (SISCISR) is set when a break is detected. The Break Detected bit (UBRKD) remains set until it is cleared by the software. The Receiving Break bit (RBRKD) is automatically cleared when a frame is received that is not a break.

The status of the next reception data to be read is set to the Overrun Error bit (UOER), Parity Error bit (UPER), Framing Error bit (UFER), and the Receive Break bit (RBRKD). Each of these statuses is updated when reception data is read from the Receive FIFO Register (SIRFIFOn).

During DMA transfer, an error is signaled and DMA transfer stops with error data remaining in the Receive FIFO if either an error (Framing Error, Parity Error, or Overrun Error) or a Reception time out (TOUT) is detected. If a Reception Error occurs during DMA transfer, use the Receive FIFO Reset bit (RFRST) of the FIFO Control Register (SIFCRn) to clear the Receive FIFO. However, a software reset will be required if a reception overrun error has occurred. Refer to “11.3.10 Software Reset” for more information.

11.3.9 Reception Time Out

A Reception time out is detected and the Reception Time Out bit (TOUT) of the DMA/Interrupt Status Register (SIDISR) is set under the following conditions.

- Non-DMA transfer mode (SIDICRn.RDE = 0):

  When at least 1 Byte of reception data exists in the Receive FIFO and the data reception time for the 2 frames (2 Bytes) after the last reception has elapsed

- DMA transfer mode (SIDICRn.RDE = 1):

  When the data reception time for the 2 frames (2 Bytes) after the last reception has elapsed regardless of whether reception data exists in the Receive FIFO

11.3.10 Software Reset

It is necessary to reset the FIFO and perform a software reset in the following situations.

1. After transmission data is set in FIFO, etc., transmission started but stopped before its completion
2. An overrun occurred during data reception

Software reset is performed by setting the Software Reset bit (SWRST) of the FIFO Control Register (SIFCR). This bit automatically returns to “0” after initialization is complete. This bit must be set again since all SIO registers are initialized by software resets.

11.3.11 Error Detection/Interrupt Signaling

An interrupt is signaled if an error or an interrupt cause is detected, the corresponding status bit is set and the corresponding Interrupt Enable bit is set.

The following figure shows the relationship between the status bit for each interrupt cause and each interrupt enable bit. Please refer to the explanation for each status bit for more information about each interrupt cause.

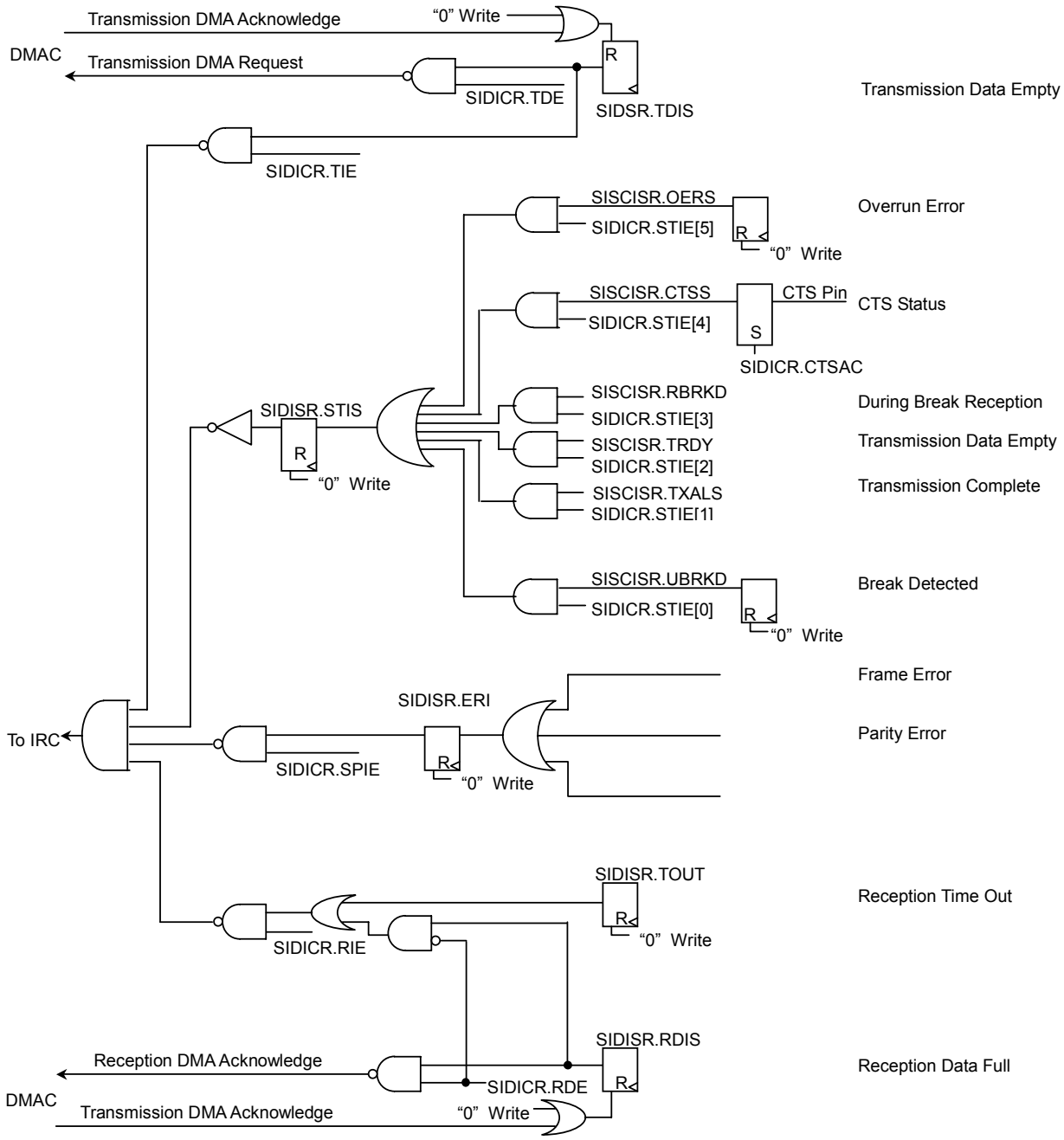


Figure 11.3.3 Relationship Between Interrupt Status Bits and Interrupt Signals

11.3.12 Multi-Controller System

The Multi-Controller System consists of one Master Controller, and multiple Slave Controllers as shown below in Figure 11.3.4.

In the case of the Multi-Controller System, the Master Controller transmits an address (ID) frame to all Slave Controllers, then transmits and receives data with the selected Slave Controller. Slave Controllers that were not selected will ignore this data.

Data frames whose data frame Wake Up bits (WUB) are "1" are handled as address (ID) frames. Data frames whose Wake Up bit (WUB) is "0" are handled as data frames.

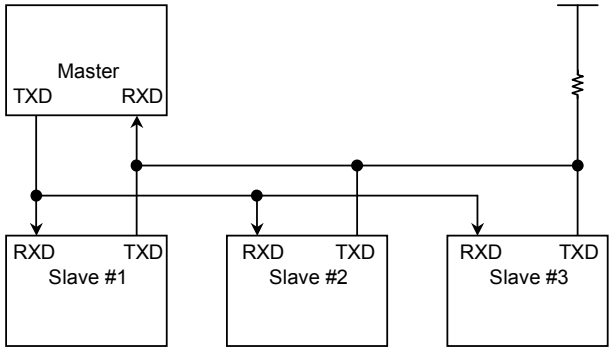


Figure 11.3.4 Example Configuration of Multi-Controller System

The data transfer procedure for the Multi-Controller System is as follows.

(1) The Master and Slave Controllers set the Mode field (UMODE) of the Line Control Register (SILCR) to "10" or "11" to set the Multi-Controller System mode. Also, the Slave Controller sets the open drain enable bit (UODE) of the Line Control Register (SILCR), setting the TXD output signal to open drain output.

(2) The Slave Controller sets the Reception Wake Up bit (RWUB) of the Line Control Register (SILCR), making it possible to receive address (ID) frames from the Master Controller.

(3) The Master Controller sets the Transmission Wake Up bit (TWUB) of the Line Control Register (SILCR), and transmits the address (ID) of the selected Slave Controller. This causes the address (ID) frame to be transmitted. The Reception after Address Transmission Wake Up bit (RWUB) is cleared, enabling reception of data frames.

(4) Since the Reception Wake Up bit (RWUB) is set, the Slave Controller generates an interrupt to the CPU by receiving an address (ID) frame. The CPU compares its own address (ID) and the received data together. If they do not match, the Reception Wake Up bit (RWUB) is cleared, making data frame reception possible.

(5) The Master Controller and the selected Slave Controller clear the Transmission Wake Up bit (TWUB) of the Line Control Register (SILCR), then set the mode that transmits data frames.

(6) Transmit/Receive data between the Master Controller and the selected Slave Controller. Then, Slave Controllers that were not selected ignore data frames since the Reception Wake Up bit (RWUB) is still set.

11.4 Registers

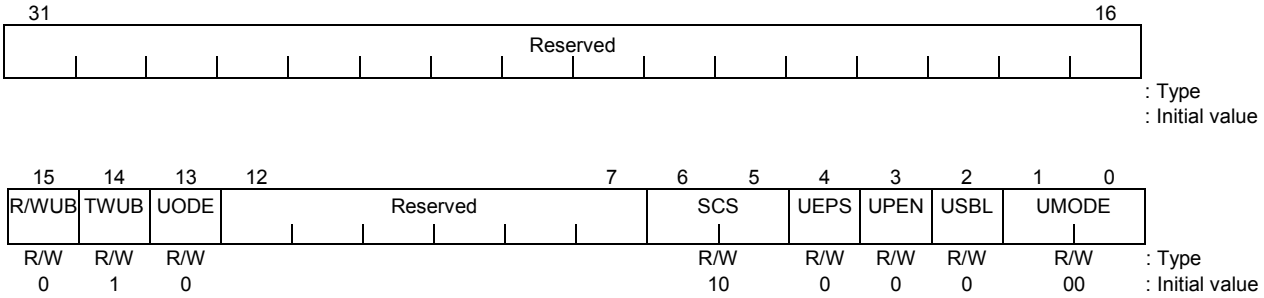
With the exception of DMA access to the Transmit FIFO Register or the Receive FIFO Register, please use Word access when accessing register in the Serial I/O Port.

Table 11.4.1 SIO Registers

Offset Address	Mnemonic	Register Name
SIO0 (Channel 0)		
0xF300	SILCR0	Line Control Register 0
0xF304	SIDICR0	DMA/Interrupt Control Register 0
0xF308	SIDISR0	DMA/Interrupt Status Register 0
0xF30C	SISCISR0	Status Change Interrupt Status Register 0
0xF310	SIFCR0	FIFO Control Register 0
0xF314	SIFLCR0	Flow Control Register 0
0xF318	SIBGR0	Baud Rate Control Register 0
0xF31C	SITFIFO0	Transmit FIFO Register 0
0xF320	SIRFIFO0	Receive FIFO Register 0
SIO1 (Channel 1)		
0xF400	SILCR1	Line Control Register 1
0xF404	SIDICR1	DMA/Interrupt Control Register 1
0xF408	SIDISR1	DMA/Interrupt Status Register 1
0xF40C	SISCISR1	Status Change Interrupt Status Register 1
0xF410	SIFCR1	FIFO Control Register 1
0xF414	SIFLCR1	Flow Control Register 1
0xF418	SIBGR1	Baud Rate Control Register 1
0xF41C	SITFIFO1	Transmit FIFO Register 1
0xF420	SIRFIFO1	Receive FIFO Register 1

11.4.1 Line Control Register 0 (SILCR0) 0xF300 (Ch. 0)
Line Control Register 1 (SILCR1) 0xF400 (Ch. 1)

These registers specify the format of asynchronous transmission/reception data.



Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	RWUB	Receive Wake Up Bit	Wake Up Bit for Receive (Default: 0) When in the Multi-Controller System mode, this field selects whether to receive address (ID) frames whose Wake Up bits (WUB) are “1” or to receive data frames whose Wake Up bits (WUB) are “0”. This value is undefined when not in the Multi-Controller System mode. 0: Receive data frames. 1: Receive address (ID) frames.	R/W
14	TWUB	Transmit Wake Up Bit	Wake Up Bit for Transmit (Default: 1) When in the Multi-Controller System mode, this field specifies the Wake Up bit (WUB). This value is undefined when not in the Multi-Controller System mode. 0: Data frame transfer (WUB = 0) 1: Address (ID) frame transfer (WUB = 1)	R/W
13	UODE	Open Drain Enable	TXD Open Drain Enable (Default: 0) This field selects the output mode of the TXD signal. When in the Multi-Controller System mode, the Slave Controller must set the TXD signal to Open Drain. 0: Totem pole output 1: Open drain output	R/W
12:7		Reserved		—
6:5	SCS	Clock Select	SIO Clock Select (Default: 00) This field selects the serial transfer clock. The clock frequency that is the serial transfer clock divided by 16 becomes the baud rate (bps). 00: Internal clock (IMBUSCLK) 01: Baud rate generator output that divided IMBUSCLK 10: External clock (SCLK) 11: Baud rate generator output that divided SCLK	R/W
4	UEPS	Even Parity Select	UART Even Parity Select (Default: 0) This field selects the parity mode. 0: Odd parity 1: Even parity	R/W
3	UPEN	Parity Check Enable	UART Parity Enable (Default: 0) This field selects whether to perform the parity check. This bit must be cleared in multidrop systems (i.e., when the UMODE field is 10 or 11.) 0: Disable the parity check 1: Enable the parity check	R/W

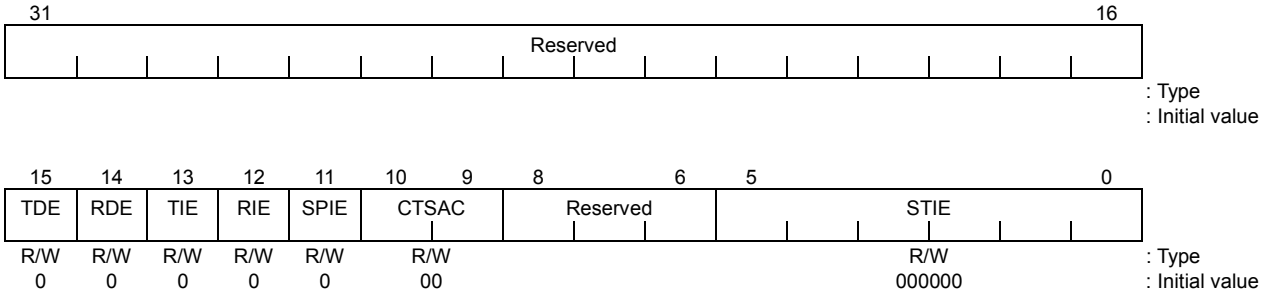
Figure 11.4.1 Line Control Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
2	USBL	Stop Bit Length	UART Stop Bit Length (Default: 0) This field specifies the stop bit length. 0: 1 bit 1: 2 bit	R/W
1:0	UMODE	Mode	UART Mode (Default: 00) This field sets the data frame mode. 00: 8-bit data length 01: 7-bit data length 10: Multi-Controller 8-bit data length 11: Multi-Controller 7-bit data length	R/W

Figure 11.4.1 Line Control Register (2/2)

11.4.2 DMA/Interrupt Control Register 0 (SIDICR0) 0xF304 (Ch. 0)
DMA/Interrupt Control Register 1 (SIDICR1) 0xF404 (Ch. 1)

These registers use either DMA or interrupts to execute the Host Interface.


31
16
Reserved
: Type
: Initial value
15 14 13 12 11 10 9 8 6 5 0
TDE RDE TIE RIE SPIE CTSAC Reserved STIE
R/W R/W R/W R/W R/W R/W R/W
0 0 0 0 0 00 000000
: Type
: Initial value

Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	TDE	Transmit DMA Transfer Enable	Transmit DMA Enable (Default: 0) This field sets whether to use DMA in the method for writing transmission data to the Transmit FIFO. 0: Do not use DMA. 1: Use DMA.	R/W
14	RDE	Receive DMA Transfer Enable	Receive DMA Enable (Default: 0) This field sets whether to use DMA in the method for reading reception data from the Receive FIFO. 0: Do not use DMA. 1: Use DMA.	R/W
13	TIE	Transmit Data Empty Interrupt Enable	Transmit Data Empty Interrupt Enable (Default: 0) When there is open space in the Transmit FIFO, this field sets whether to signal an interrupt. Set “0” when in the DMA Transmit mode (TDE = 1). 0: Do not signal an interrupt when there is open space in the Transmit FIFO. 1: Signal an interrupt when there is open space in the Transmit FIFO.	R/W
12	RIE	Reception Data Full Interrupt Enable	Receive Data Full Interrupt Enable (Default: 0) This field sets whether to signal interrupts when reception data is full (SIDISRn.RDIS = 1) or a reception time out (SIDISRn.TOUT = 1) occurs. Set to “0” when in the DMA Receive mode (RDE = 1). 0: Do not signal interrupts when reception data is full/reception time out occurred. 1: Signal interrupts when reception data is full/reception time out occurred.	R/W
11	SPIE	Reception Error Interrupt Enable	Receive Data Error Interrupt Enable (Default: 0) This field sets whether to signal interrupts when a reception error (Frame Error, Parity Error, Overrun Error) occurs (SIDISR.ERI = 1). 0: Do not signal reception error interrupts. 1: Signal reception error interrupts.	R/W
10:9	CTSAC	CTSS Active Condition	CTSS Active Condition (Default: 00) This field specifies status change interrupt request conditions using the CTS Status (CTSS) of the Status Change Interrupt Status Register. 00: Do not detect CTS signal changes. 01: Rising edge of the CTS pin 10: Falling edge of the CTS pin 11: Both edges of the CTS pin	R/W
8:6		Reserved		—

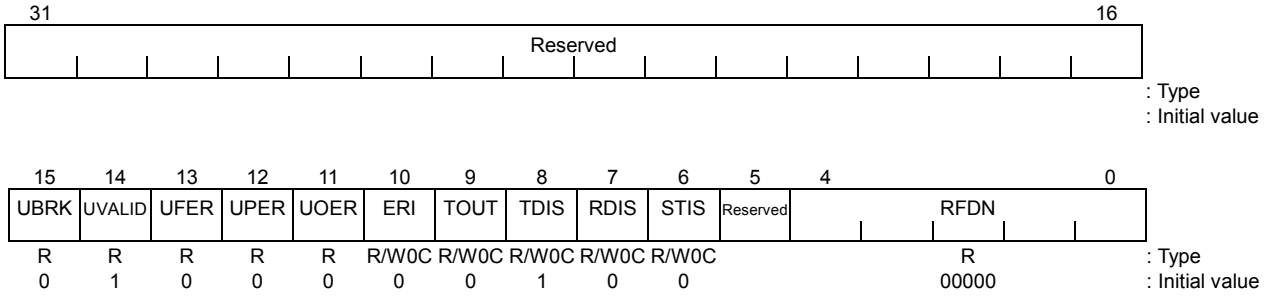
Figure 11.4.2 DMA/Interrupt Control Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
5:0	STIE	Status Change Interrupt Enable	Status Change Interrupt Enable (Default: 0x00) This field sets the set conditions of the Status Change bit (STIS) of the DMA/Interrupt Status Register (SIDISR). The condition is selected depending on which bit of the Status Change Interrupt Status Register (SISCISR) is set. (Multiple selections are possible.) An SIO interrupt is asserted when STIC is “1”. 000000: Do not detect status changes. 1*****: Set “1” to STIS when the Overrun bit (OERS) is “1”. *1****: Set “1” to STIS when a change occurs in a condition set by the CTSS Active Condition field (CTSAC) in the CTS Status bit (CTSS). **1***: Set “1” to STIS when the Break bit (RBRKD) becomes “1”. ***1**: Set “1” to STIS when the Transmit Data Empty bit (TRDY) becomes “1”. ****1*: Set “1” to STIS when the Transmission Complete bit (TXALS) becomes “1”. *****1: Set “1” to STIS when the Break Detection bit (UBRKD) becomes “1”.	R/W

Figure 11.4.2 DMA/Interrupt Control Register (2/2)

11.4.3 DMA/Interrupt Status Register 0 (SIDISR0) 0xF308 (Ch. 0)
DMA/Interrupt Status Register 1 (SIDISR1) 0xF408 (Ch. 1)

These registers indicate the DMA or interrupt status information.


31
16
Reserved
: Type
: Initial value
15 14 13 12 11 10 9 8 7 6 5 4 0
UBRK UVALID UFER UPER UOER ERI TOUT TDIS RDIS STIS Reserved RFDN
R R R R R R/W0C R/W0C R/W0C R/W0C R/W0C R
0 1 0 0 0 0 0 1 0 0 00000
: Type
: Initial value

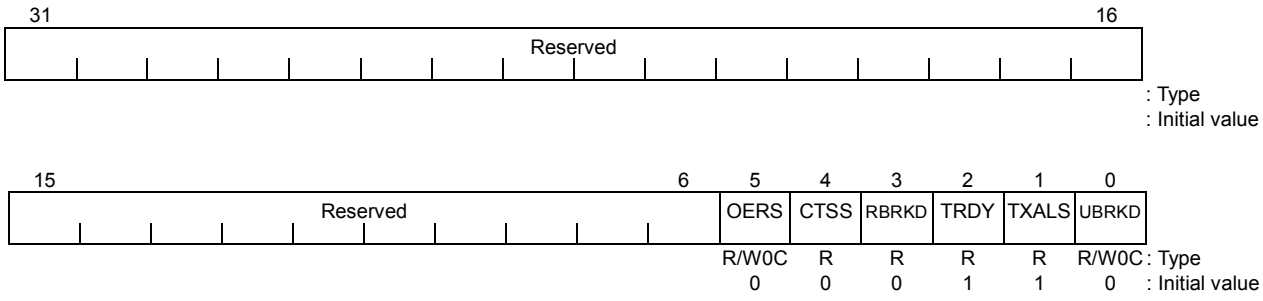
Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	UBRK	Receive Break	UART Break (Default: 0) This field indicates the break reception status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: No breaks 1: Detect breaks	R
14	UVALID	Receive FIFO Available Status	UART Available Data (Default: 1) This field indicates whether or not data exists in the Receive FIFO (SIRFIFO). 0: Data exists in the Receive FIFO. 1: No data exists in the Receive FIFO.	R
13	UFER	Frame Error	UART Frame Error (Default: 0) This field indicates the frame error status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no frame errors. 1: There are frame errors.	R
12	UPER	Parity Error	UART Parity Error (Default: 0) This field indicates the parity error status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no parity errors. 1: There are parity errors.	R
11	UOER	Overrun Error	UART Overrun Error (Default: 0) This register indicates the overrun status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no overrun errors. 1: There are overrun errors.	R
10	ERI	Reception Error Interrupt	Receive Data Error Interrupt (Default: 0) This bit is immediately set to “1” when a reception error (Frame Error, Parity Error, or Overrun Error) is detected.	R/W0C
9	TOUT	Reception Time Out	Time Out (Default: 0) This bit is set to “1” when a reception time out occurs.	R/W0C
8	TDIS	Transmission Data Empty	Transmit DMA/Interrupt Status (Default: 1) This bit is set when available space of the amount set by the Transmit FIFO Request Trigger Level (TDIL) of the FIFO Control Register (SIFCR) exists in the Transmit FIFO.	R/W0C

Figure 11.4.3 DMA/Interrupt Status Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
7	RDIS	Reception Data Full	Receive DMA/Interrupt Status (Default: 0) This bit is set when valid data of the amount set by the Receive FIFO Request Trigger Level (RDIL) of the FIFO Control register (SIFCR) is stored in the Receive FIFO.	R/W0C
6	STIS	Status Change	Status Change Interrupt Status (Default: 0) This bit is set when at least one of the interrupt statuses selected by the Status Change Interrupt Condition field (STIE) of the DMA/Interrupt Control Register (SIDICR) becomes “1”.	R/W0C
5		Reserved		—
4:0	RFDN	Reception Data Stage Status	Receive FIFO Data Number (Default: 00000) This field indicates how many stages of reception data remain in the Receive FIFO (0 – 16 stages).	R

Figure 11.4.3 DMA/Interrupt Status Register (2/2)

11.4.4 Status Change Interrupt Status Register 0 (SISCISR0) 0xF30C (Ch. 0)
Status Change Interrupt Status Register 1 (SISCISR1) 0xF40C (Ch. 1)

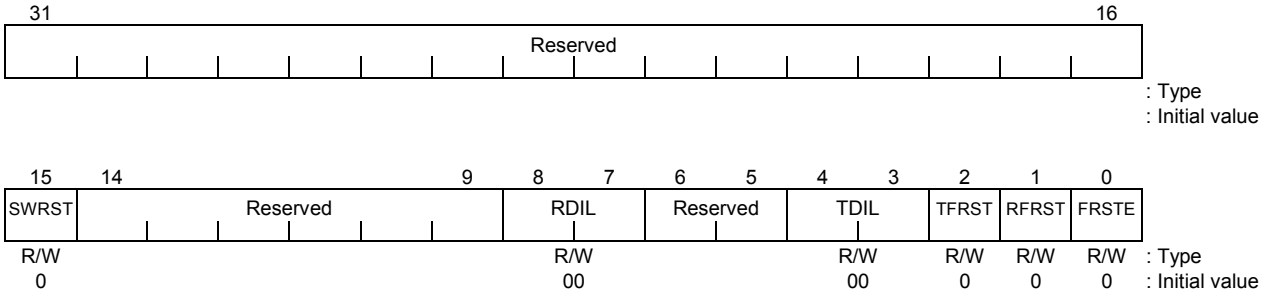


Bit	Mnemonic	Field Name	Description	Read/Write
31:6	—	Reserved	—	—
5	OERS	Overrun Error	Overrun Error Status (Default: 0) This bit is immediately set to “1” when an overrun error is detected. This bit is cleared when a “0” is written.	R/W0C
4	CTSS	CTS Status	CTS Terminal Status (Default: 0) This field indicates the status of the CTS signal. 1: The CTS signal is High. 0: The CTS signal is Low.	R
3	RBRKD	Receiving Break	Receive Break (Default: 0) This bit is set when a break is detected. This bit is automatically cleared when a frame that is not a break is received. 1: Current status is Break. 0: Current status is not Break.	R
2	TRDY	Transmission Data Empty	Transmit Ready (Default: 1) This bit is set to “1” if at least one stage in the Transmit FIFO is free.	R
1	TXALS	Transmission Complete	Transmit All Sent (Default: 1) This bit is set to “1” if the Transmit FIFO and all transmission shift registers are empty.	R
0	UBRKD	Break Detected	UART Break Detect (Default: 0) This bit is set when a break is detected. Once set, this bit remains set until cleared by writing a “0” to it.	R/W0C

Figure 11.4.4 Status Change Interrupt Status Register

11.4.5 FIFO Control Register 0 (SIFCR0) 0xF310 (Ch. 0)
FIFO Control Register 1 (SIFCR1) 0xF410 (Ch. 1)

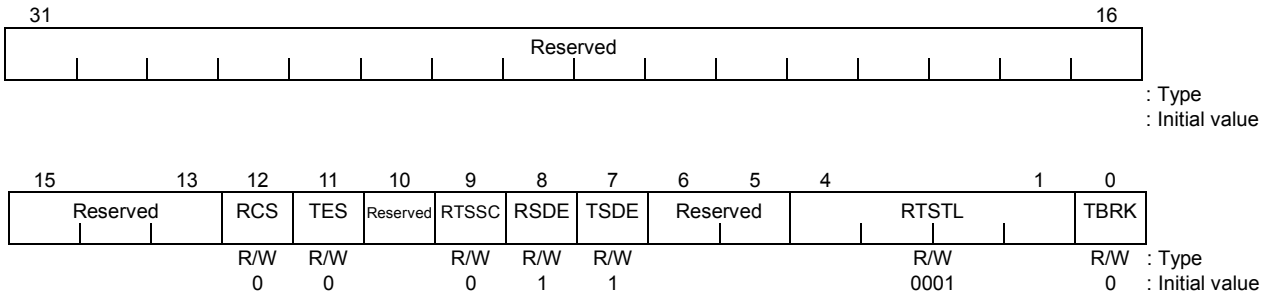
These registers set control of the Transmit/Receive FIFO buffer.



Bit	Mnemonic	Field Name	Description	Read/Write
31:16	—	Reserved	—	—
15	SWRST	Software Reset	Software Reset (Default: 0) This field performs SIO resets except for the FIFOs. Setting this bit to “1” initiates the reset. Set registers are also initialized. This bit returns to “0” when initialization is complete. 0: Normal operation 1: SIO software reset	R/W
14:9	—	Reserved	—	—
8:7	RDIL	Receive FIFO Request Trigger Level	Receive FIFO DMA/Interrupt Trigger Level (Default: 00) This register sets the level for reception data transfer from the Receive FIFO. 00: 1 Byte 01: 4 Bytes 10: 8 Bytes 11: 12 Bytes	R/W
6:5	—	Reserved	—	—
4:3	TDIL	Transmit FIFO Request Trigger Level	Transmit FIFO DMA/Interrupt Trigger Level (Default: 00) This register sets the level for transmission data transfer to the Transmit FIFO. 00: 1 Byte 01: 4 Bytes 10: 8 Bytes 11: Setting disabled	R/W
2	TFRST	Transmit FIFO Reset	Transmit FIFO Reset (Default: 0) The Transmit FIFO buffer is reset when this bit is set. This bit is valid when the FIFO Reset Enable bit (FRSTE) is set. Cancel reset by using the software to clear this bit. 0: During operation 1: Reset Transmit FIFO	R/W
1	RFRST	Receive FIFO Reset	Receive FIFO Reset (Default: 0) The Receive FIFO buffer is reset when this bit is set. This bit is valid when the FIFO Reset Enable bit (FRSTE) is set. Cancel reset by using the software to clear this bit. 0: During operation 1: Reset Receive FIFO	R/W
0	FRSTE	FIFO Reset Enable	FIFO Reset Enable (Default: 0) This field is the Reset Enable for the Transmit/Receive FIFO buffer. The FIFO is reset by combining the Transmit FIFO Reset bit (TFRST) and Receive FIFO Reset bit (RFRST). 0: During operation 1: Reset Enable	R/W

Figure 11.4.5 FIFO Control Register

11.4.6 Flow Control Register 0 (SIFLCR0) 0xF314 (Ch. 0)
Flow Control Register 1 (SIFLCR1) 0xF414 (Ch. 1)

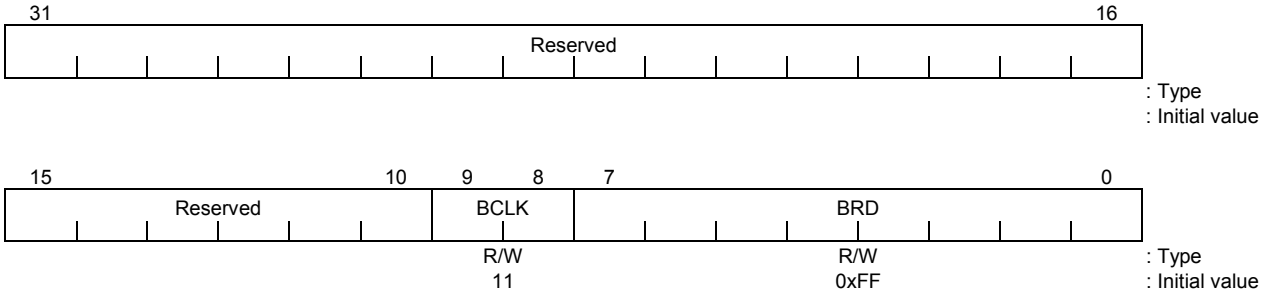


Bit	Mnemonic	Field Name	Description	Read/Write
31:13		Reserved		—
12	RCS	RTS Signal Control Select	RTS Control Select (Default: 0) This field sets the reception flow control using RTS output signals. 0: Disable flow control using RTS signals. 1: Enable flow control using RTS signals.	R/W
11	TES	CTS Signal Control Select	CTS Control Select (Default: 0) This field sets the transmission flow control using CTS input signals. 0: Disable flow control using CTS signals. 1: Enable flow control using CTS signals.	R/W
10		Reserved		—
9	RTSSC	RTS Software Control	RTS Software Control (Default: 0) This register is used for software control of RTS output signals. 0: Set the RTS signal to Low (can receive data). 1: Sets the RTS signal to High (transmission pause request)	R/W
8	RSDE	Serial Data Reception Disable	Receive Serial Data Disable (Default: 1) This is the Serial Data Disable bit. When this bit is cleared, data reception starts after the start bit is detected. The RTS signal will not become High even if this bit is cleared. 0: Enable (can receive data) 1: Disable (halt reception)	R/W
7	TSDE	Serial Data Transmit Disable	Transmit Serial Data Disable (Default: 1) This is the Serial Data Transmission Disable bit. When this bit is cleared, data transmission starts. When set, transmission stops after completing transmission of the current frame. 0: Enable (can transmit data) 1: Disable (halt transmission)	R/W
6:5		Reserved		—
4:1	RTSTL	RTS Active Trigger Level	RTS Trigger Level (Default: 0001) The RTS hardware control assert level is set by the reception data stage count of the Receive FIFO. 0000: Disable setting 0001: 1 : 1111: 15	R/W
0	TBRK	Break Transmission	Break Transmit (Default: 0) Transmits a break. The TXD signal is Low while TBRK is set to “1”. 0: Disable (clear break) 1: Enable (transmit break)	R/W

Figure 11.4.6 Flow Control Register

11.4.7 Baud Rate Control Register 0 (SIBGR0) 0xF318 (Ch. 0)
Baud Rate Control Register 1 (SIBGR1) 0xF418 (Ch. 1)

These registers select the clock that is provided to the baud rate generator and set the divide value.



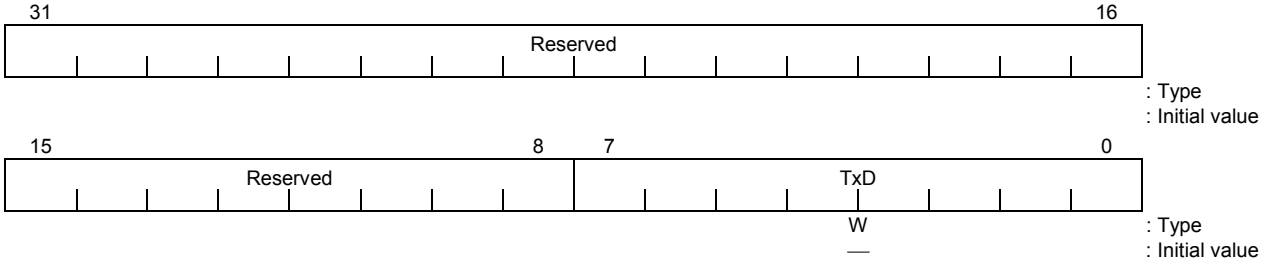
Bit	Mnemonic	Field Name	Description	Read/Write
31:10		Reserved		—
9:8	BCLK	Baud Rate Generator Clock	Baud Rate Generator Clock (Default: 11) This field sets the input clock for the baud rate generator. 00: Select prescalar output T0 (fc/2) 01: Select prescalar output T2 (fc/8) 10: Select prescalar output T4 (fc/32) 11: Select prescalar output T6 (fc/128)	R/W
7:0	BRD	Baud Rate Divide Value	Baud Rate Divide Value (Default: 0xFF) This field set divide value BRG of the baud rate generator. This value is expressed as a binary value.	R/W

Figure 11.4.7 Baud Rate Control Register

11.4.8 Transmit FIFO Register 0 (SITFIFO0) 0xF31C (Ch. 0)
Transmit FIFO Register 1 (SITFIFO1) 0xF41C (Ch. 1)

When using the DMA Controller to perform DMA transmission, set the following addresses in the Destination Address Register (DMDARn) of the DMA Controller according to the Endian Mode bit (DMCCRn.LE) setting of the DMA Controller.

- Little Endian: 0xF31C (Ch.0), 0xF41C (Ch.1)
- Big Endian: 0xF31F (Ch.0), 0xF41F (Ch.1)



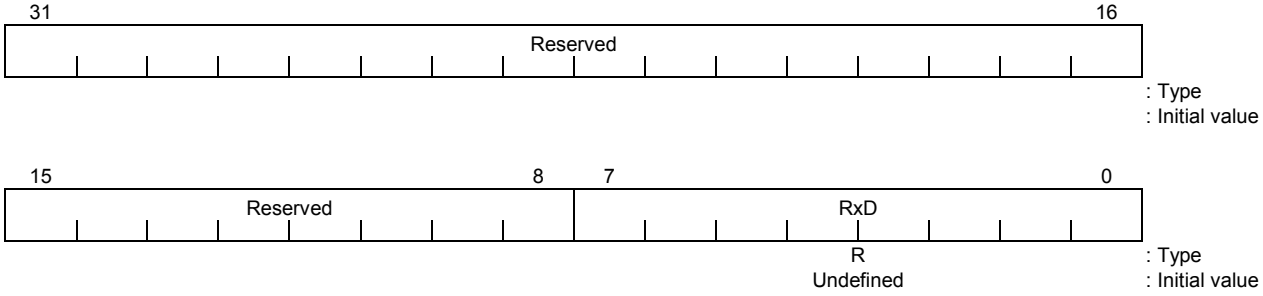
Bit	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7:0	TxD	Transmission Data	Transmit Data Data written to this register are written to the Transmit FIFO.	W

Figure 11.4.8 Transmit FIFO Register

11.4.9 Receive FIFO Register 0 (SIRFIFO0) 0xF320 (Ch. 0)
Receive FIFO Register 1 (SIRFIFO1) 0xF420 (Ch. 1)

When using the DMA Controller to perform DMA transmission, set the following addresses in the Destination Address Register (DMDARn) of the DMA Controller according to the Endian Mode bit (DMCCRn.LE) setting of the DMA Controller.

- Little Endian: 0xF320 (Ch.0), 0xF420 (Ch.1)
- Big Endian: 0xF323 (Ch.0), 0xF423 (Ch.1)



Bit	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7:0	RxD	Reception Data	Receive Data This field reads reception data from the Receive FIFO. Reading this register updates the Reception Data Status.	R

Figure 11.4.9 Receive FIFO Register

12. Timer/Counter

12.1 Features

The TX4938 has an on-chip 3-channel timer/counter.

- 32-bit Up Counter: 3 Channels
- Interval Timer Mode (Channel 0, 1, 2)
- Pulse Generator Mode (Channel 0, 1)
- Watchdog Timer Mode (Channel 2)
- Timer Output Signal (TIMER[1:0]) × 2
- Counter Input Signal (TCLK): × 1
- Watchdog Timer Reset Output (WDRST*): × 1

12.2 Block Diagram

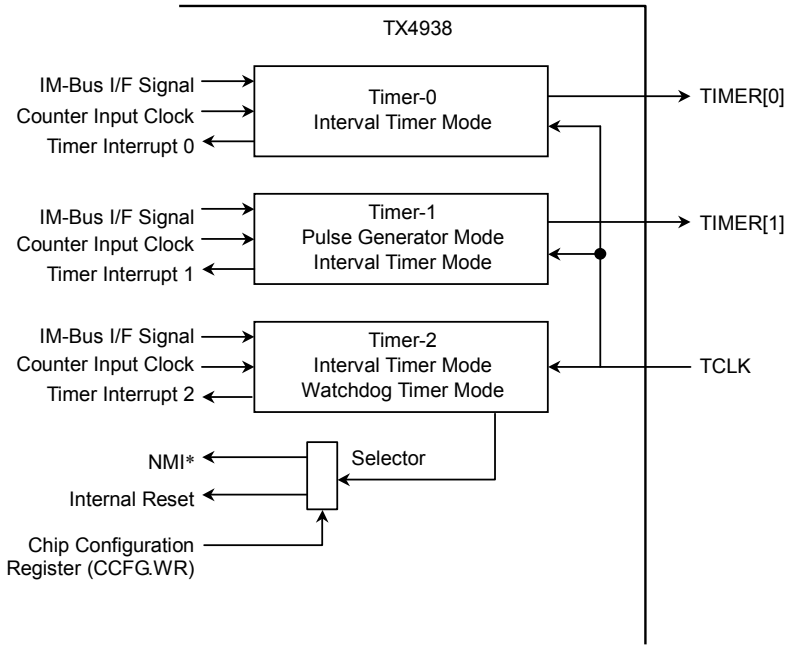


Figure 12.2.1 Connecting Timer Module Inside the TX4938

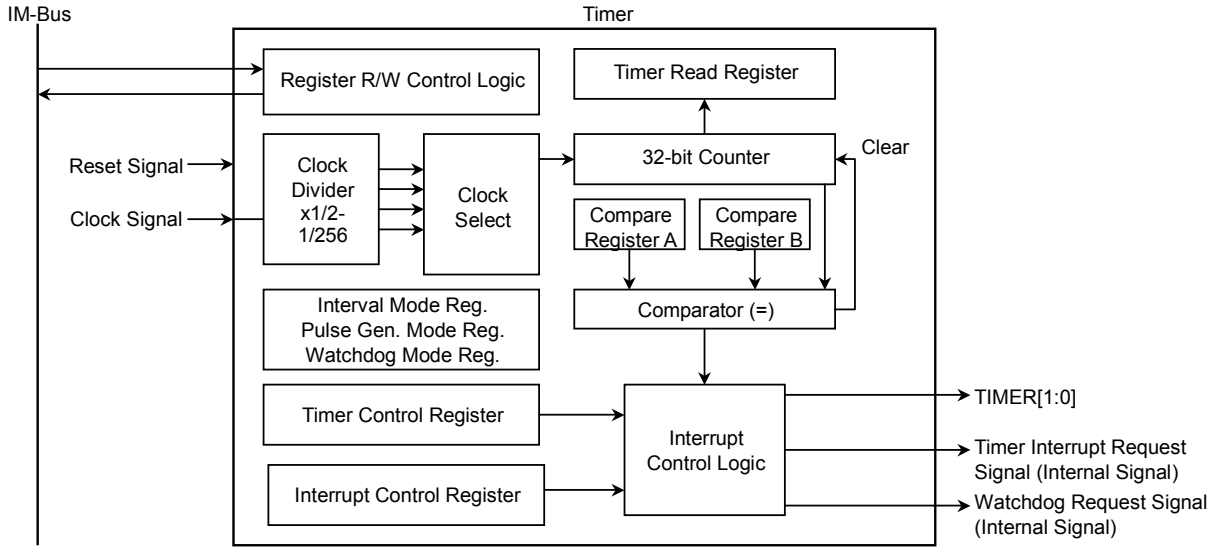


Figure 12.2.2 Timer Internal Block Diagram

12.3 Detailed Explanation

12.3.1 Overview

The TX4938 has an on-chip 3-channel 32-bit timer/counter. Each channel supports the following modes.

(1) Interval Timer Mode (Timer 0, 1, 2)

This mode periodically generates interrupts.

(2) Pulse Generator Mode (Timer 0, 1)

This is the pulse signal output mode.

(3) Watchdog Timer Mode (Timer 2)

This mode is used to monitor system abnormalities.

12.3.2 Counter Clock

The clock used for counting can be set to a frequency that is 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, or 1/256 of the internal clock (IMBUSCLK) frequency, or can be selected from nine counter input signal (TCLK) types. Divide Register *n* (TMCCDRn) and the Counter Clock Select bit (TMTCRn.CCS) are used to select the counter clock. In this situation, IMBUSCLK is the internal clock signal which is the G-Bus clock divided by 2. See “Chapter 6 Clocks” for more information.

The counter input signal (TCLK) is used by three channels. Using TCLK makes it possible to count external events. The External Clock Edge bit (TMTCRn.ECES) can be used to select the clock rising/falling count.

Set the TCLK clock frequency to 45% or less of IMBUSCLK (TCLK = 27 MHz or less when IMBUSCLK = 60 MHz). The following tables shows example count times when using 60 MHz or 66 MHz IMBUSCLK.

Table 12.3.1 Divide Value and Count (IMBUSCLK = 60 MHz)

Divide Rate	TMCCDRn. CCD	Counter Clock Frequency (Hz)	Resolution (ns)	Max. Set Time (sec.)	TMCPRAn Value for 1 sec.
2	000	30.0 M	33.33	143.17	30000000
4	001	15.0 M	66.67	286.33	15000000
8	010	7.5 M	133.33	572.66	7500000
16	011	3.75 M	266.67	1145.32	3750000
32	100	1.9 M	533.33	2290.65	1875000
64	101	937.5 K	1066.67	4581.30	937500
128	110	468.8 K	2133.33	9162.60	468750
256	111	234.4 K	4266.67	18325.20	234375

Table 12.3.2 Divide Value and Count (IMBUSCLK = 66 MHz)

Divide Rate	TMCCDRn. CCD	Counter Clock Frequency (Hz)	Resolution (ns)	Max. Set Time (sec.)	TMCPRAn Value for 1 sec.
2	000	33.0 M	30.30	130.15	33000000
4	001	16.5 M	60.61	260.30	16500000
8	010	8.3 M	121.21	520.60	8250000
16	011	4.1 M	242.42	1041.20	4125000
32	100	2.1 M	484.85	2082.41	2062500
64	101	1031.3 K	969.70	4164.82	1031250
128	110	515.6 K	1939.39	8329.63	515625
256	111	257.8 K	3878.79	16659.27	257813

12.3.3 Counter

Each channel has an independent 32-bit counter. Set the Timer Count Enable bit (TMTCRn.TCE) and the 32-bit counter will start counting.

Clear the Timer Count Enable bit to stop the counter. If the Counter Reset Enable bit (TMTCRn.CRE) is set, then the counter will be cleared also. The Watchdog Timer Disable bit (TMWTRM2.WDIS) must be set in order to stop and clear this counter when in the Watch Dog Timer mode.

Also, reading the Timer Read Register (TMTRR) makes it possible to fetch the counter value.

12.3.4 Interval Timer Mode

The Interval Timer mode is used to periodically generate interrupts. Setting the Timer Mode field (TMTCRn.TMODE) of the Timer Control Register to “00” sets the timer to the Interval Timer mode. This mode can be used by all timers.

When the count value matches the value of Compare Register A (TMCPRAn), the Interval Timer TMCPRA Status bit (TMTISRn.TIIS) of the Timer Interrupt Status Register is set. When the Interval Timer Interrupt Enable bit (TMITMRn.TIIE) of the Interval Timer Mode Register is set, timer interrupts occur. When a “0” is written to the Interval Timer TMCPRA Status bit (TMTISRn.TIIS), TIIS is cleared and timer interrupts stop.

If the Timer Zero Clear Enable bit (TMITMRn.TZCE) is set, the counter is cleared to 0 if the count value matches the Compare Register A (TMCPRAn) value. Count operation stops when the Timer Zero Clear Enable bit (TMITMRn.TZCE) is cleared.

The level of the TIMER[1:0] output signal stays in the initial state (Low) in this mode. Output is undefined when changing from the Pulse Generator mode to this mode. Figure 12.3.1 shows an outline of the count operation and generation of interrupts when in the Interval Timer mode and Figure 12.3.2 shows the operation when using an external input clock.

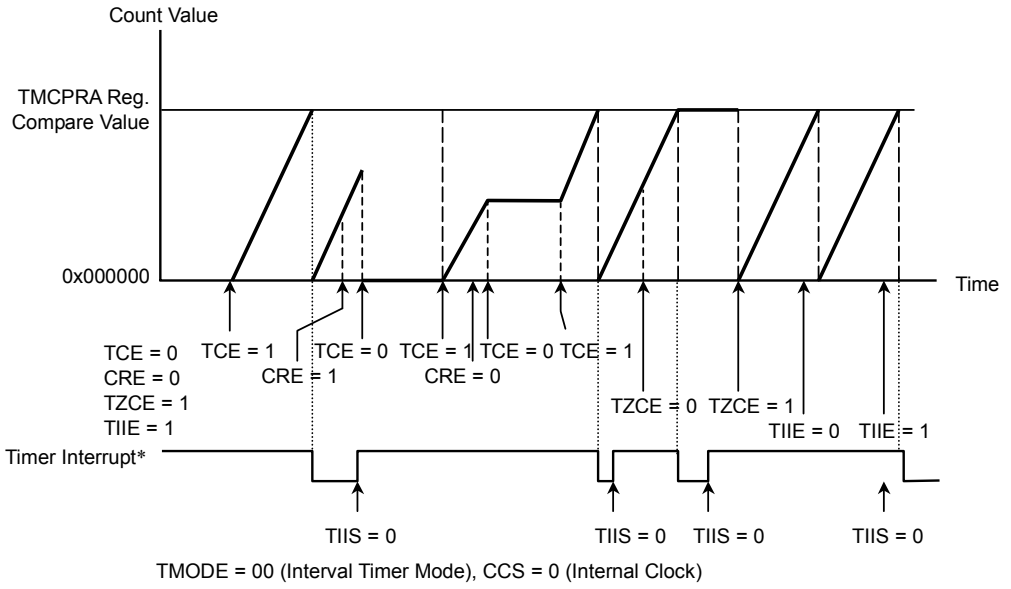


Figure 12.3.1 Operation Example of Interval Timer (Using Internal Clock)

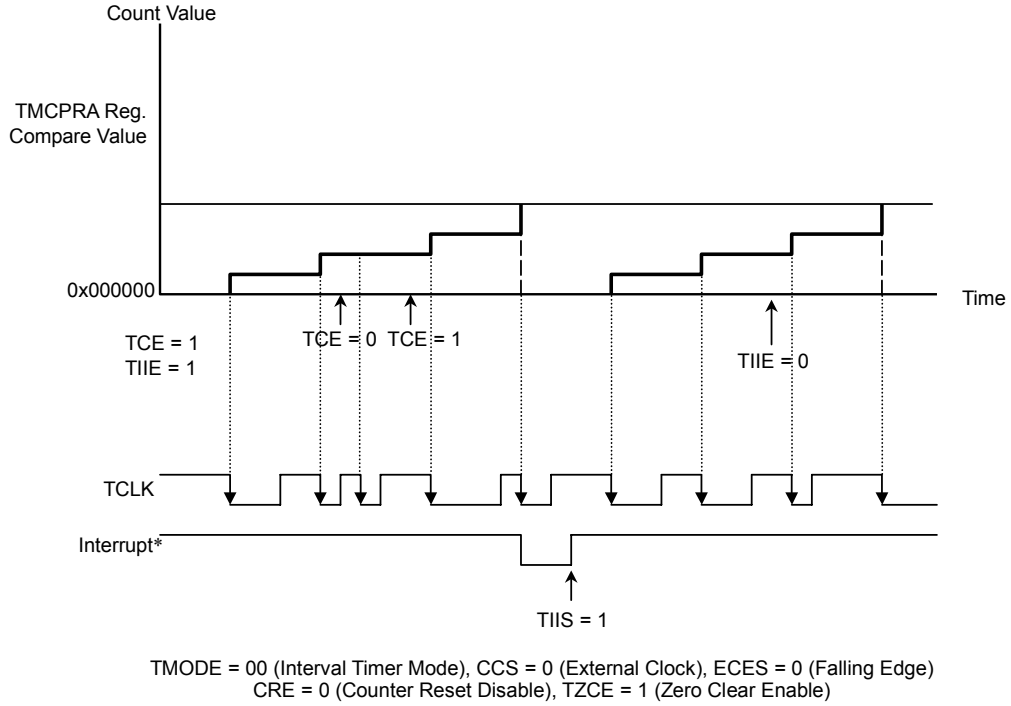


Figure 12.3.2 Operation Example of the Interval Timer (External Input Clock: Rising Edge Operation)

12.3.5 Pulse Generator Mode

When in the Pulse Generator mode, use Compare Register A (TMCPRAn) and Compare Register B (TMCPRBn) to output a particular period and particular duty square wave to the TIMER[n] signal. Setting the Timer Mode field (TMTCRn.TMODE) of the Timer Control Register to "01" sets the timer to the Pulse Generator mode. Timer 0 and Timer 1 can be used, but Timer 2 cannot.

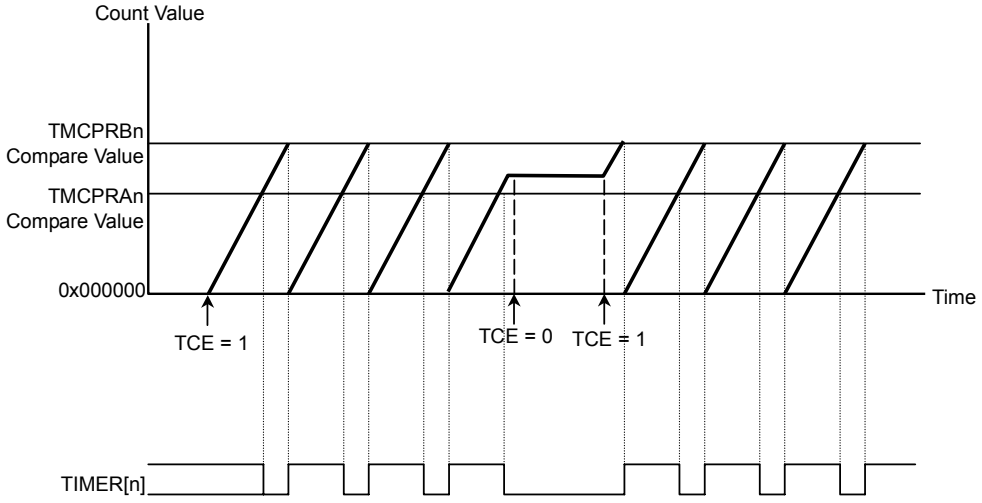
The initial state of the TIMER[n] signal can be set by the Flip Flop Default bit (TMPGMRn.FFI) of the Pulse Generator Mode Register.

The TIMER[n] output signal reverses when the counter value matches the value set in Compare Register A (TMCPRAn). The TIMER[n] output signal reverse again, clearing the counter when the counter continues counting and the value set in Compare Register B (TMCPRBn) and the counter value match. Consequently, a value greater than that in Compare Register A (TMCPRAn) must not be set in Compare Register B (TMCPRBn).

Interrupts can be generated in the Pulse Generator mode as well. However, this is not standard practice.

The Pulse Generator TMCPRA Status bit (TMTISRn.TPIAS) of the Timer Interrupt Status Register is set when the count value matches the value of Compare Register A (TMCPRAn). Timer interrupts are generated when the TMCPRA Interrupt Enable bit (TMPGMRn.TPIAE) of the Pulse Generator Mode Register is set.

Similarly, the Pulse Generator TMCPRB Status bit (TMTISRn.TPIBS) of the Timer Interrupt Status Register is set when the count value matches the value of Compare Register B (TMCPRBn). Timer interrupts are generated when the TMCPRB Interrupt Enable bit (TMPGMRn.TPIBE) of the Pulse Generator Mode Register is set.



TMODE1 = 01 (Pulse Generator Mode), CCS= 0 (Internal Clock)
FFI = 1 (Initial High), CRE = 0 (Counter Reset Disable)

Figure 12.3.3 Operation Example of the Pulse Generator Mode

12.3.6 Watchdog Timer Mode

The Watchdog Timer mode is used to monitor system anomalies. The software periodically clears the counter and judges an anomaly to exist if the counter is not cleared within a specified period of time. Then, either the TX4938 is internally reset or an NMI is signaled to the TX49/H3 core. Set the Timer mode field (TMTCR2.TMODE) of the Timer Control Register to "10" to set the timer to the Watchdog Timer mode. This mode can only be used by Timer 2.

Use the Watchdog Reset bit (WR) of the Chip Configuration Register (CCFG) to select whether to perform an internal reset or signal an NMI. Set this bit to "1" to select Watchdog Reset, or set it to "0" to select NMI Signaling.

When the timer count reaches the value programmed in Compare Register A (TMCPRA2), the Watchdog Timer TMCPRA Match Status bit in the Timer Interrupt Status Register (TMTISR2.TWIS) is set. Either the watchdog timer reset or NMI is issued if the Timer Watchdog Enable bit in the Watchdog Timer Mode Register (TMWTMR2.TWIE) is set.

When the watchdog timer reset is selected, the Watchdog Reset Status bit in the Chip Configuration Register (CCFG.WDRST) is set. If the Watchdog Reset External Output bit in the Chip Configuration Register (CCFG.WDREXEN) is cleared, the entire TX4938 is initialized but the configuration registers. Setting the Watchdog Reset External Output bit (CCFG.WDREXEN) causes the WDRST* signal to be asserted. This does not initialize the TX4938. The WDRST* signal remains asserted until the RESET* signal is asserted. Assertion of the RESET* signal deasserts the WDRST* signal and initializes the TX4938.

There are three ways of stopping NMI signaling from being performed.

1. Clear the Watchdog Timer Interrupt Status bit (TMTISR2.TWIS) of the timer Interrupt Status Register.
2. Clear the counter by writing "1" to the Watchdog Timer Clear bit (TMWTMR2.TWC) of the Watchdog Timer Mode Register.
3. Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.TWIE) while the Watchdog Timer Disable bit (TMWTMR2.WDIS) is still set.

It is possible to stop the counter when in the Watchdog Timer mode by clearing the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register while the Watchdog Timer Disable bit (TMWTMR2.WDIS) of the Watchdog Timer Mode Register is set to "1".

It is also possible to stop the counter by clearing the Counter Clock Divide Cycle Enable bit (TMTCR2.CCDE) of the Timer Control Register when the internal clock is being used as the counter clock.

It is not possible to directly write "0" to the Watchdog Timer Disable bit (TMWTMR2.WDIS). There are two ways to clear this bit.

1. Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.WDIS)
2. Clear the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register

In Watchdog Timer mode, the TIMER[1:0] outputs remain at logic high.

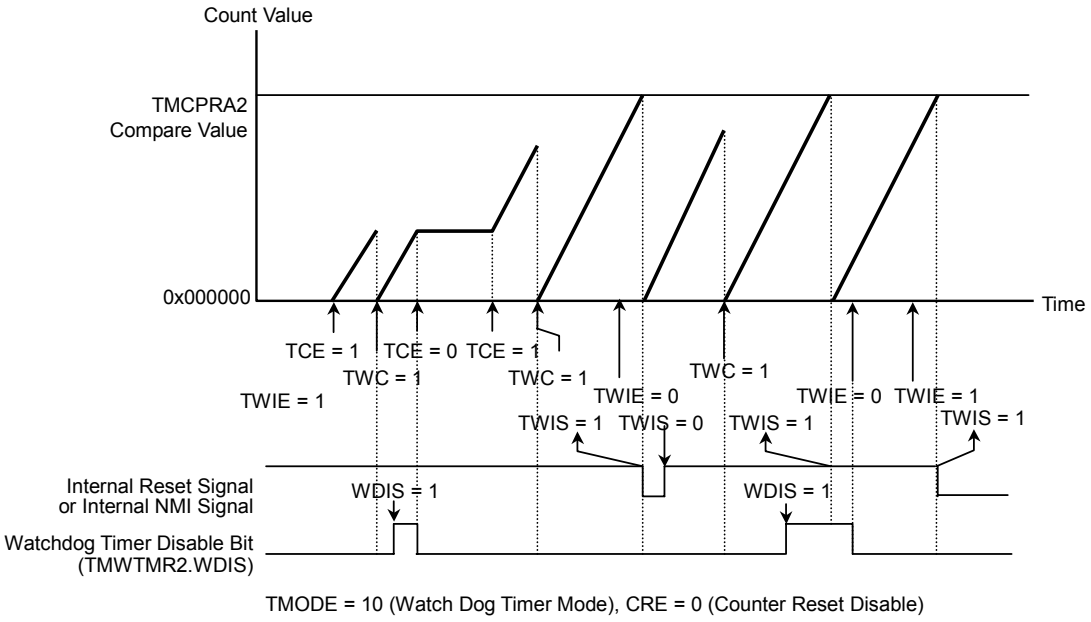


Figure 12.3.4 Operation Example of the Watchdog Timer Mode

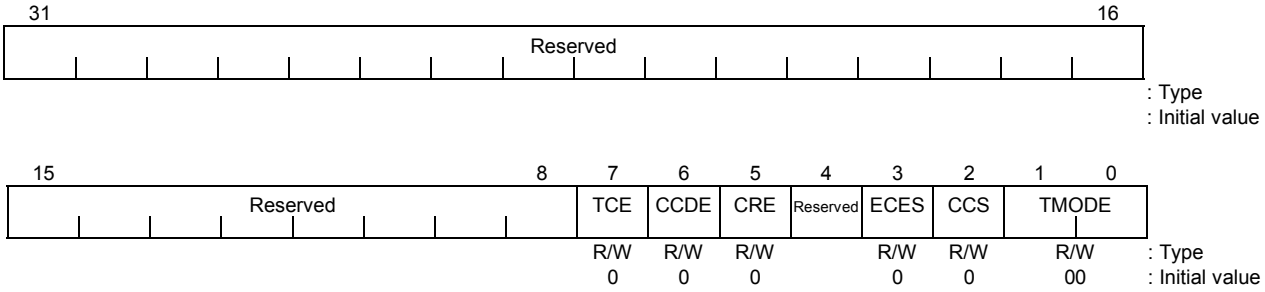
12.4 Registers

Table 12.4.1 Timer Register List

Offset Address	Register Symbol	Register Name
Time 0 (TMR0)		
0xF000	TMTCR0	Timer Control Register 0
0xF004	TMTISR0	Timer Interrupt Status Register 0
0xF008	TMCPRA0	Compare Register A 0
0xF00C	TMCPRB0	Compare Register B 0
0xF010	TMITMR0	Interval Timer Mode Register 0
0xF020	TMCCDR0	Divide Cycle Register 0
0xF030	TMPGMR0	Pulse Generator Mode Register 0
0xF040	TMWTMR0	(Reserved)
0xF0F0	TMTRR0	Timer Read Register 0
Timer 1 (TMR1)		
0xF100	TMTCR1	Timer Control Register 1
0xF104	TMTISR1	Timer Interrupt Status Register 1
0xF108	TMCPRA1	Compare Register A 1
0xF10C	TMCPRB1	Compare Register B 1
0xF110	TMITMR1	Interval Timer Mode Register 1
0xF120	TMCCDR1	Divide Cycle Register 1
0xF130	TMPGMR1	Pulse Generator Mode Register 1
0xF140	TMWTMR1	(Reserved)
0xF1F0	TMTRR1	Timer Read Register 1
Timer 2 (TMR2)		
0xF200	TMTCR2	Timer Control Register 2
0xF204	TMTISR2	Timer Interrupt Status Register 2
0xF208	TMCPRA2	Compare Register A 2
0xF20C	TMCPRB2	(Reserved)
0xF210	TMITMR2	Interval Timer Mode Register 2
0xF220	TMCCDR2	Divide Cycle Register 2
0xF230	TMPGMR2	(Reserved)
0xF240	TMWTMR2	Watchdog Timer Mode Register 2
0xF2F0	TMTRR2	Timer Read Register 2

12.4.1 Timer Control Register *n* (TMTCRn)

TMTCR0 0xF000
TMTCR1 0xF100
TMTCR2 0xF200



Bit	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7	TCE	Timer Counter Enable	Timer Count Enable (Default: 0) This field controls whether the counter runs or stops. When in the Watchdog mode, counter operation only stops when the Watchdog Timer Disable bit (TMWTMR2.WDIS) of the Watchdog Timer Mode Register is set. When the Watchdog Timer Disable bit is cleared, the value of this Timer Count Enable bit becomes “0”, but the count continues. 0: Stop counter (the counter is also cleared to “0” when CRE = 1) 1: Counter operation	R/W
6	CCDE	Counter Clock Divider Enable	Counter Clock Divide Enable (Default: 0) This bit enables the divide operation of the internal clock (IMBUSCLK). The counter stops if this bit is set to “0” when the internal bus clock is in use. 0: Disable 1: Enable	R/W
5	CRE	Counter Reset Enable	Counter Reset Enable (Default: 0) This bit controls the counter reset when the TCE bit was used to stop the counter. During CRE = 1, reset the counter if TCE is set from 1 to 0. During TCE = 0, the counter isn’t reset if CRE is set from 0 to 1. When TCE = 1 and CRE = 0, stop and reset the counter if TCE is set to 0 and CRE is set to 1 simultaneously. 1: Stop and reset the counter to “0” when the TCE bit is cleared to “0”. 0: Only stop the counter when the TCE bit is cleared to “0”.	R/W
4		Reserved		—
3	ECES	External Clock Edge Select	External Clock Edge Select (Default: 0) This bit specifies the counter operation edge when using the counter input signal (TCLK). 0: Falling edge of the counter input signal (TCLK) 1: Rising edge of the counter input signal (TCLK)	R/W
2	CCS	Counter Clock Select	Counter Clock Select (Default: 0) This bit specifies the timer clock. 0: Internal clock (IMBUSCLK) 1: External input clock (TCLK)	R/W
1:0	TMODE	Timer Mode	Timer Mode (Default: 00) This bit specifies the timer operation mode. 11: Reserved 10: Watchdog Timer mode (Timer 2), Reserved (Timer 0, 1) 01: Pulse Generator mode (Timer 0, 1), Reserved (Timer 2) 00: Interval Timer mode	R/W

Figure 12.4.1 Timer Control Register

12.4.2 Timer Interrupt Status Register *n* (TMTISRn)

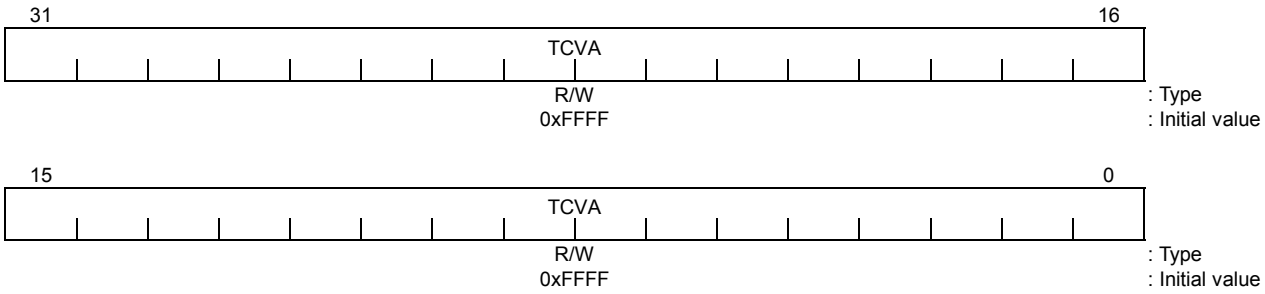
TMTISR0 0xF004
TMTISR1 0xF104
TMTISR2 0xF204

Bit	Mnemonic	Field Name	Description	Read/Write
31:4		Reserved		—
3	TWIS	Watchdog Timer Status	Watchdog Timer TMCPRA Match Status (Default: 0) (This bit is Reserved in the case of the TMTISR0 Register and the TMTISR1 Register.) When in the Watchdog Timer mode, this bit is set when the counter value matches Compare Register 2 (TMCPRA2). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Negate interrupt 1: Invalid	R/W0C
2	TPIBS	Pulse Generator TMCPRB Status	Pulse Generator TMCPRB Match Status (Default: 0) (This bit is Reserved in the case of the TMTISR2 Register.) When in the Pulse Generator mode, this bit is set when the counter value matches Compare Register Bn (TMCPRBn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid	R/W0C
1	TPIAS	Pulse Generator TMCPRA Status	Pulse Generator TMCPRA Match Status (Default: 0) (This bit is Reserved in the case of the TMTISR2 Register.) When in the Pulse Generator mode, this bit is set when the counter value matches Compare Register A n (TMCPRAn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid	R/W0C
0	TIIS	Interval Timer TMCPRA Status	Interval Timer TMCPRA Match Status (Default: 0) When in the Interval Timer mode, this bit is set when the counter value matches Compare Register A n (TMCPRAn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid	R/W0C

Figure 12.4.2 Timer Interrupt Status Register

12.4.3 Compare Register An (TMCPRAn)

TMCPRA0 0xF008
TMCPRA1 0xF108
TMCPRA2 0xF208

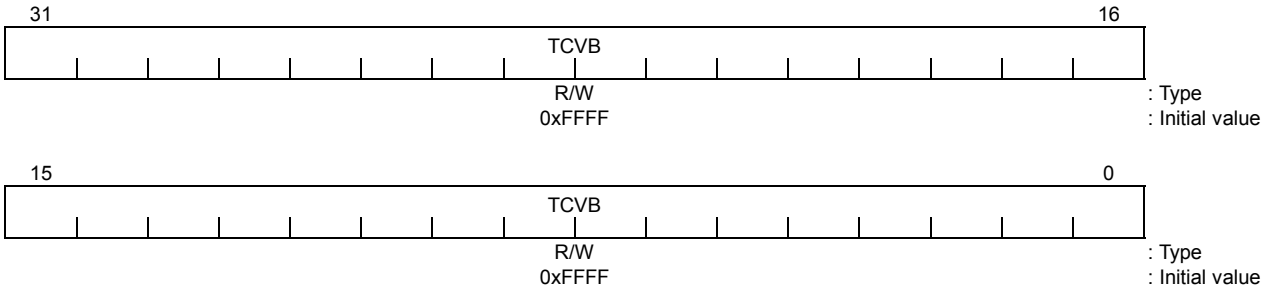


Bits	Mnemonic	Field Name	Description	Read/Write
31:0	TCVA	Timer Compare Register A	Timer Compare Value A (Default: 0xFFFFFFFF) Sets the timer compare value as a 32-bit value. This register can be used in all modes.	R/W

Figure 12.4.3 Compare Register A

12.4.4 Compare Register Bn (TMCPRBn)

TMCPRB0 0xF00C
TMCPRB1 0xF10C



Bits	Mnemonic	Field Name	Description	Read/Write
31:0	TCVB	Timer Compare Value B	Timer Compare Value B (Default: 0xFFFFFFFF) Sets the timer compare value as a 32-bit value. This register can only be used when in the Pulse Generator mode. Please set a value greater than that in Compare Register A.	R/W

Figure 12.4.4 Compare Register B

12.4.5 Interval Timer Mode Register *n* (TMITMRn)

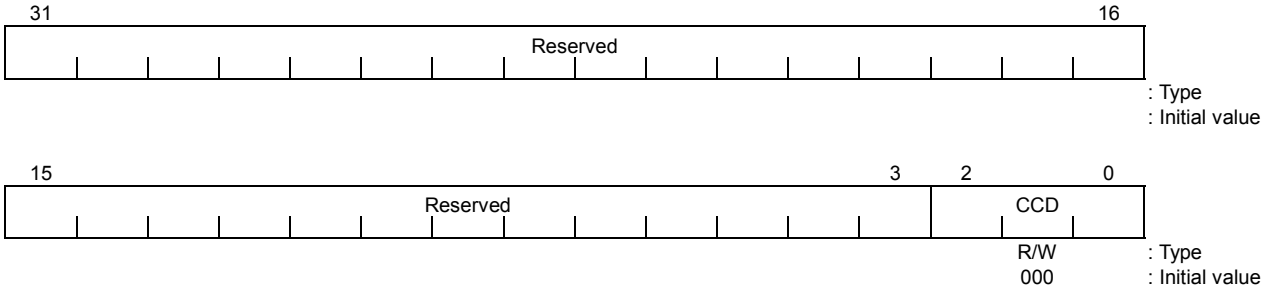
TMITMR0 0xF010
TMITMR1 0xF110
TMITMR2 0xF210

Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	TIIE	Interval Timer Interrupt Enable	Timer Interval Interrupt Enable (Default: 0) Sets Interval Timer TMCPRA Interrupt Enable/Disable. 0: Disable (mask) 1: Enable	R/W
14:1		Reserved		—
0	TZCE	Interval Timer Clear Enable	Interval Timer Zero Clear Enable (Default: 0) This bit specifies whether or not to clear the counter to “0” after the count value matches Compare Register A. Count stops at this value if it is not cleared. 0: Do not clear 1: Clear	R/W

Figure 12.4.5 Interval Timer Mode Register

12.4.6 Divide Register *n* (TMCCDRn)

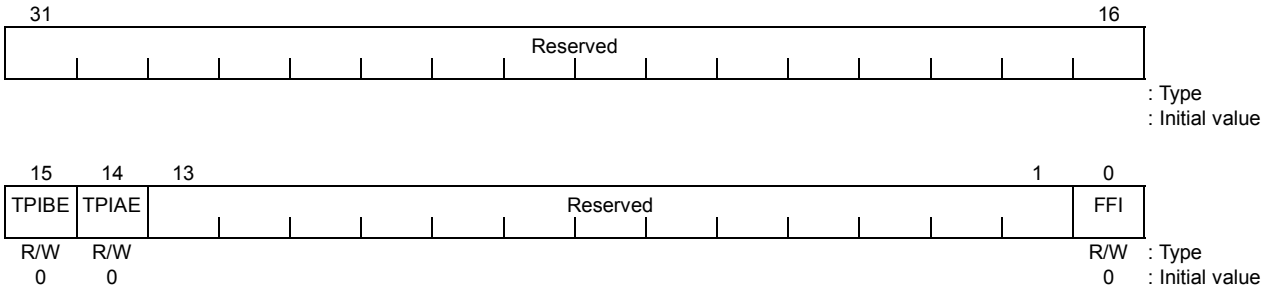
TMCCDR0 0xF020
TMCCDR1 0xF120
TMCCDR2 0xF220



Bits	Mnemonic	Field Name	Description	Read/Write
31:3		Reserved		—
2:0	CCD	Counter Clock Divide Value	Counter Clock Divide (Default: 000) These bits specify the divide value when using the internal clock (IMBUSCLK) as the counter input clock source. The binary value *n* is divided by 2^{n+1}. 000: Divide by 2^1 (f/2) 001: Divide by 2^2 (f/4) 010: Divide by 2^3 (f/8) 011: Divide by 2^4 (f/16) 100: Divide by 2^5 (f/32) 101: Divide by 2^6 (f/64) 110: Divide by 2^7 (f/128) 111: Divide by 2^8 (f/256)	R/W

Figure 12.4.6 Divide Register

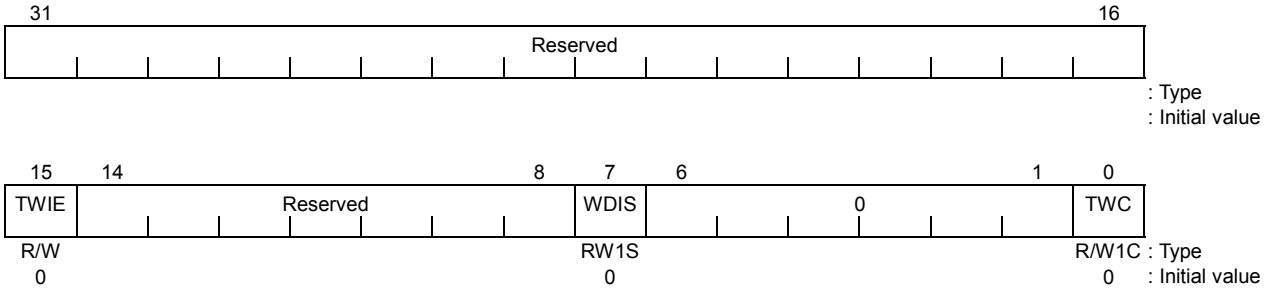
12.4.7 Pulse Generator Mode Register *n* (TMPGMRn) TMPGMR0 0xF000
TMPGMR1 0xF130



Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	TPIBE	TMCPRB Interrupt Enable	Timer Pulse Generator Interrupt by TMCPRB Enable (Default: 0) When in the Pulse Generator mode, this bit sets Interrupt Enable/Disable for when TMCPRB and the counter value match. 0: Mask 1: Do not mask	R/W
14	TPIAE	TMCPRA Interrupt Enable	Timer Pulse Generator Interrupt by TMCPRA Enable (Default: 0) When in the Pulse Generator mode, this bit sets Interrupt Enable/Disable for when TMCPRA and the counter value match. 0: Mask 1: Do not mask	R/W
13:1		Reserved		—
0	FFI	Flip Flop Default	Initial TIMER Output Level (Default: 0) This bit specifies the TIMER[n] signal default when in the Pulse Generator mode. 0: Low 1: High	R/W

Figure 12.4.7 Pulse Generator Mode Register

12.4.8 Watchdog Timer Mode Register *n* (TMWTMRn) TMWTMR2 0xF240

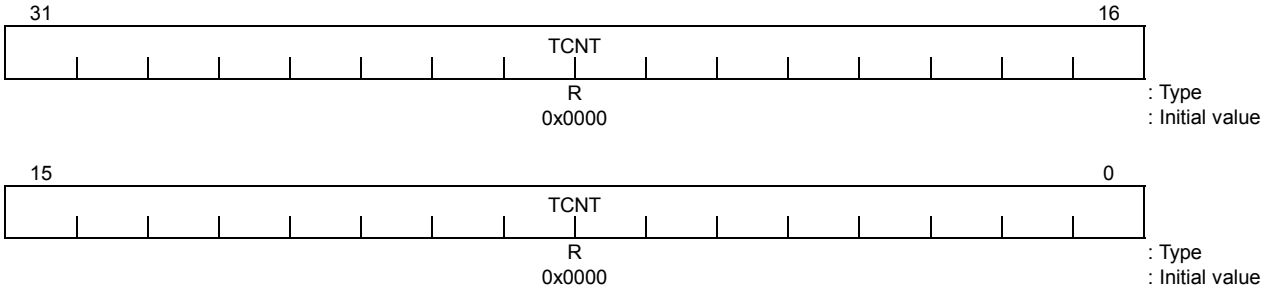


Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	TWIE	Watchdog Timer Signaling Enable	Timer Watchdog Enable (Default: 0) This bit sets NMI signaling enable/disable either when in the Watchdog Timer mode or during a reset. This bit cannot be cleared when the Watchdog Timer Disable bit (WDIS) is “0”. 0: Disable (mask) 1: Enable	R/W
14:8		Reserved		—
7	WDIS	Watchdog Timer Disable	Watchdog Timer Disable (Default: 0) Only when this bit is set can the counter be stopped by clearing the Watchdog Timer Signaling Enable bit (TWIE) or by clearing the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register. Writing “0” to this bit is not valid. This bit can be cleared in either of the following ways. Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.TWIE). Clear the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register.	R/W1S
6:1		Reserved		—
0	TWC	Watchdog Timer Clear	Watchdog Timer Clear (Default: 0) Setting this bit to “1” clears the counter. Writing “0” to this bit is not valid. This bit is always read as “0”.	R/W1C

Figure 12.4.8 Watchdog Timer Mode Register

12.4.9 Timer Read Register *n* (TMTRRn) 0xF0F0

TMTRR0 0xF0F0
TMTRR1 0xF1F0
TMTRR2 0xF2F0



Bits	Mnemonic	Field Name	Description	Read/Write
31:0	TCNT	Timer Counter	Timer Counter (Default: 0x00000000) This Read Only register is a 32-bit counter. Operation when this register is written to is undefined.	R

Figure 12.4.9 Timer Read Register 0

13. Parallel I/O Port

13.1 Characteristics

The TX4938 on-chip Parallel I/O port (PIO) is a 16-bit general-purpose parallel port. The input/output direction and the port type during output (totem pole output/open drain output) can be set for each bit.

13.2 Block Diagram

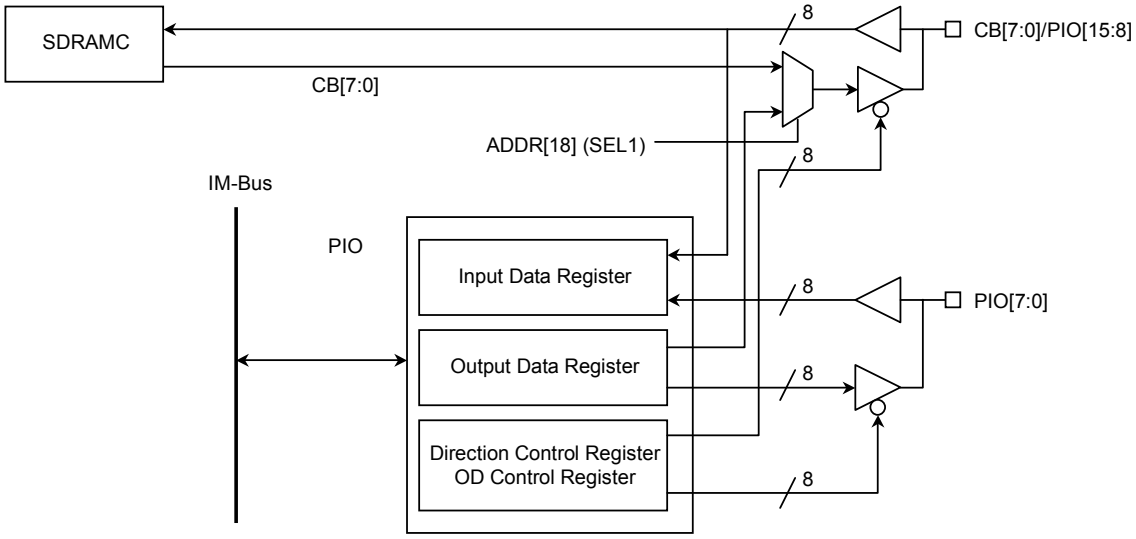


Figure 13.2.1 Parallel I/O Block Diagram

13.3 Detailed Description

13.3.1 Selecting PIO Pins

Of the 16-bit PIO signals, signals PIO[15:8] can be used in combination with 8-bit ECC check bit signals. The configuration signal (ADDR[18]) at boot up determines which function will be used. See "3.2 Boot Configuration" for more information.

13.3.2 General-purpose Parallel Port

The four following registers are used to control the PIO port.

- PIO Output Data Register (PIODO)
- PIO Input Data Register (PIODI)
- PIO Direction Control Register (PIODIR)
- PIO Open Drain Control Register (PIOOD)

PIO signals can be selected by the PIO Direction Control Register (PIODIR) for each bit as either input or output.

Signals selected as output signals output the values written into the PIO Data Output Register (PIODO). The PIO Open Drain Control Register (PIOOD) can select whether each bit is either an open drain output or a totem pole output.

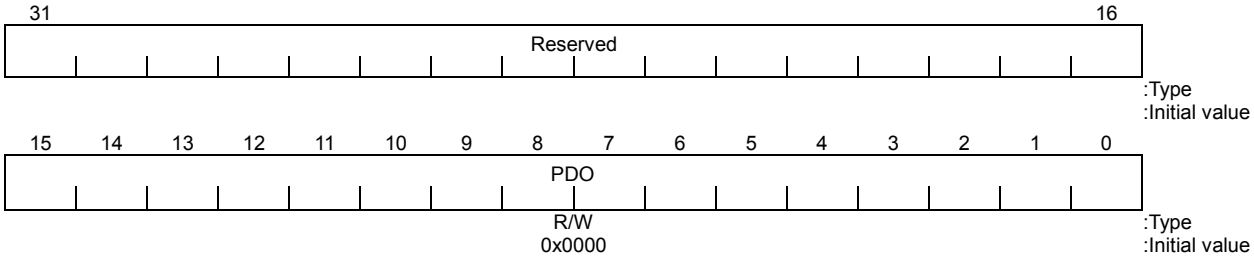
PIO signal status is indicated by the PIO Data Input Register. This register can be read out at any time regardless of the pin direction settings.

13.4 Registers

Table 13.4.1 PIO Register Map

Offset Address	Mnemonic	Register Name
0xF500	PIODO	Output Data Register
0xF504	PIODI	Input Data Register
0xF508	PIODIR	Direction Control Register
0xF50C	PIOOD	Open Drain Control Register

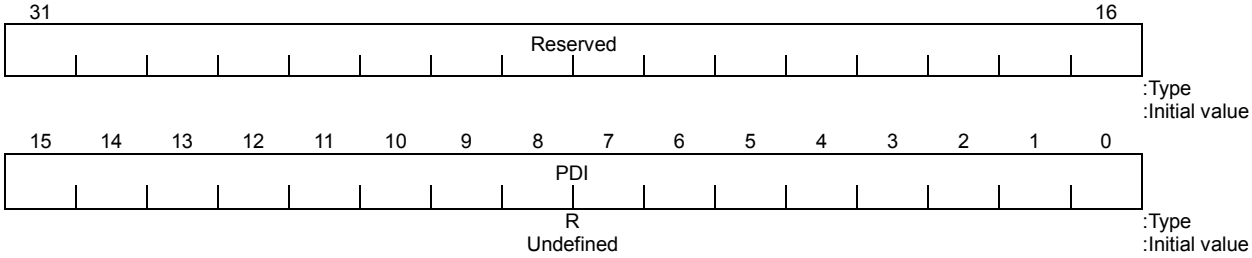
13.4.1 PIO Output Data Register (PIODO) 0xF500



Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15 :0	PDO [15:0]	Data Out	Port Data Output [15:0] (Initial value:0x0000) Data that is output to the PIO pin (PIO [15:0]).	R/W

Figure 13.4.1 PIO Output Data Register

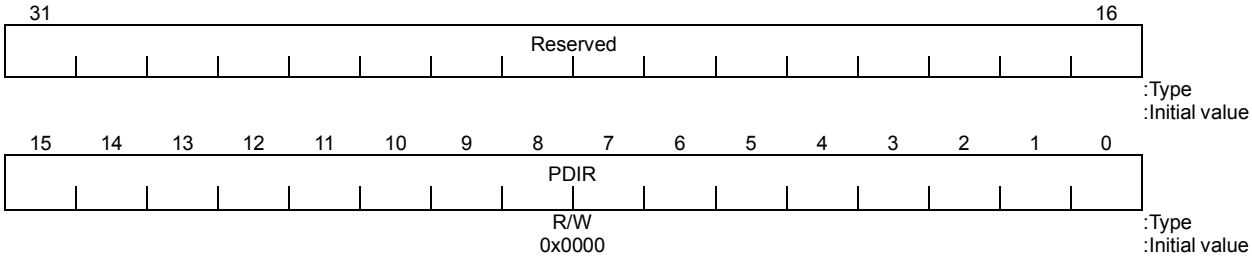
13.4.2 PIO Input Data Register (PIODI) 0xF504



Bit	Mnemonic	Field Name	Description	
31:16		Reserved		—
15 :0	PDI [15:0]	Data In	Port Data Input [15:0] (Initial value:TBD) Data that is input to the PIO pin (PIO [15:0]).	R

Figure 13.4.2 PIO Input Data Register

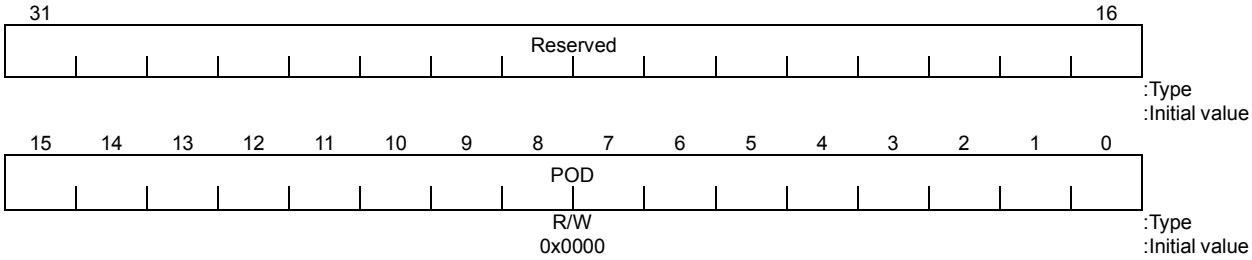
13.4.3 PIO Direction Control Register (PIODIR) 0xF508



Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15 :0	PDIR [15:0]	Direction Control	Port Direction Control [15:0] (Initial value: 0x0000) Sets the I/O direction of the PIO pin (PIO [15:0]). 0: Input (Reset) 1: Output	R/W

Figure 13.4.3 PIO Direction Control Register

13.4.4 PIO Open Drain Control Register (XPIOOD) 0xF50C



Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15 :0	POD [15:0]	Open Drain Control	Port Open Drain Control [15:0] (Initial value: 0x0000) Sets whether to use the PIO pin (PIO [15:0]) as an open drain. 0: Open drain (Reset) 1: Totem pole	R/W

Figure 13.4.4 PIO Open Drain Control Register

14. AC-link Controller

14.1 Features

ACLC, AC-link controller module can be connected to audio and/or modem CODECs described in the "Audio CODEC '97 Revision 2.1" (AC'97) defined by Intel and can operate them. Refer to the following Web site for more information regarding the AC'97 specification.

http://developer.intel.com/ial/scalableplatforms/audio/

Its features are summarized as follows.

- Up to two CODECs are supported.
- AC'97 compliant CODEC register access protocol is supported.
- CODEC register access completion is recognized by polling or interrupt.
- Recording and playback of 16-bit PCM Left&Right channels are supported.
- Recording can be selected from PCM L&R or Mic.
- Playback of 16-bit Surround, Center, and LFE channels is supported.
- Variable Rate Audio recording is supported.
- Variable Rate Audio playback is supported.
- Line 1 and GPIO slots for Modem CODEC are supported.
- AC-link low-power mode, wake-up, and warm-reset are supported
- Sample-data I/O via DMA transfer is supported.

14.2 Configuration

Figure 14.2.1 illustrates the ACLC configuration.

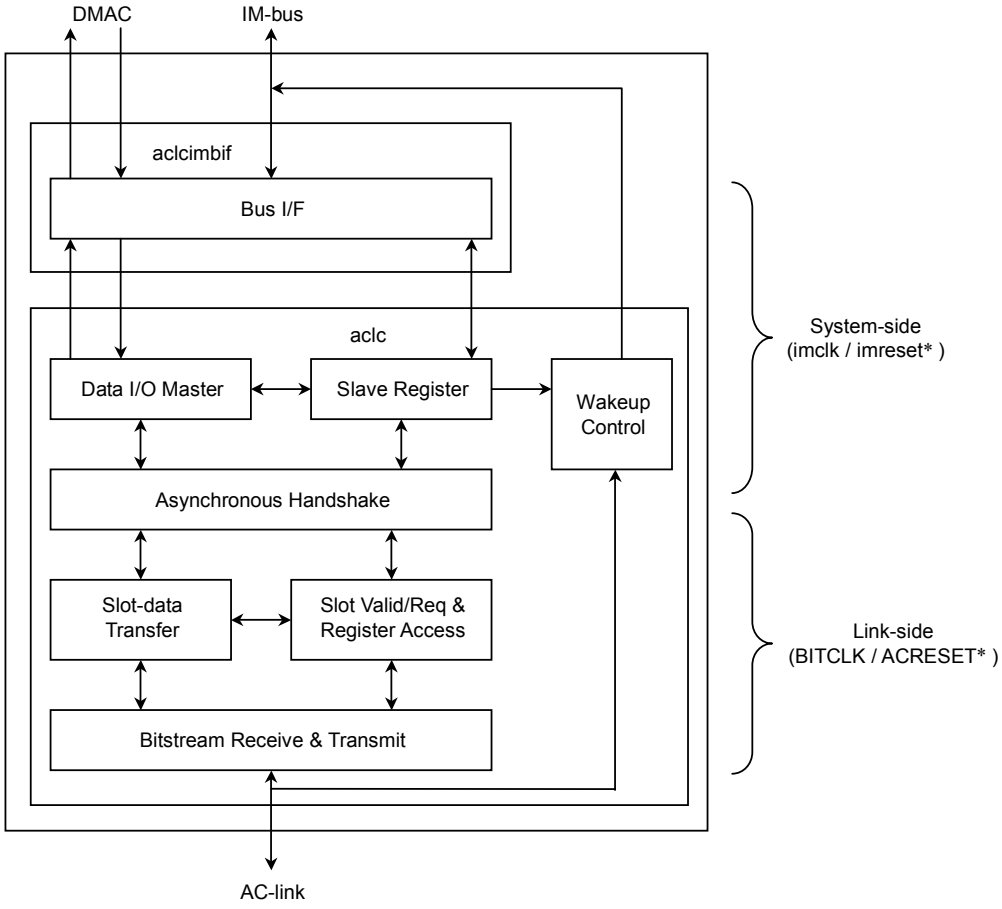


Figure 14.2.1 ACLC Module Configuration

14.3 Functional Description

ACLC provides four mechanisms to operate AC'97-compliant CODEC(s):

- AC-link status control (start-up and low-power mode)
- CODEC register access
- Sample-data transmission and reception
- GPIO operation

This section first describes the CODEC connection, chip configuration, and overall usage-flow. Then AC-link start-up sequence and the other mechanisms will be described. Using low-power mode comes last.

14.3.1 CODEC Connection

The ACLC module has two SDIN (named as SDATA_IN in the AC'97 specification) signals and supports up to two CODECs to be connected. This section shows some system configuration diagrams for typical usages. Note that the diagrams shown here is intended to provide conceptual understanding and some components may be necessary on the actual circuit board to ensure proper electrical signals. The diagrams assume CODECs compliant with the CODEC ID strapping recommendation described in the section D.5.2 of the AC'97 revision 2.1 specification.

14.3.1.1 Stereo Audio and Optional Modem Connection

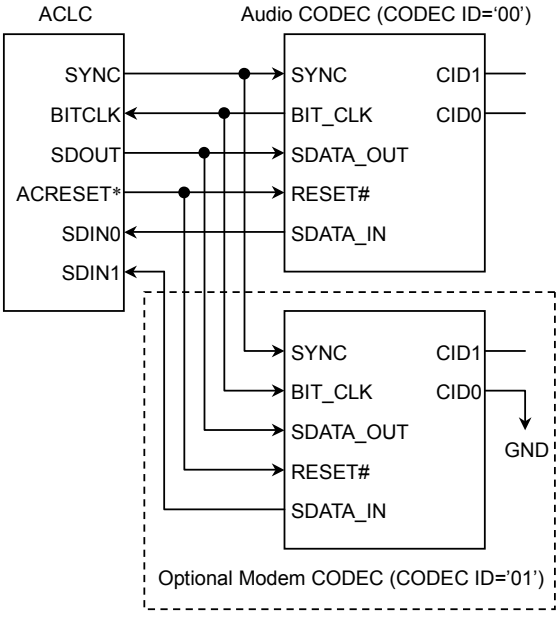


Figure 14.3.1 Stereo Audio and Optional Modem Connection Diagram

14.3.1.2 5.1 Channel Audio Connection

This sample assumes one CODEC with four DACs mapped to stereo front (3&4) and stereo rear (7&8) slots, and another CODEC with two DACs mapped to center (6) and LFE (9) slots.

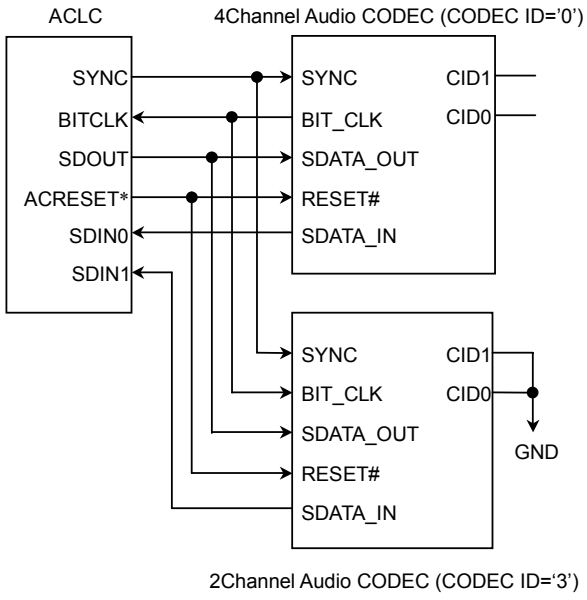


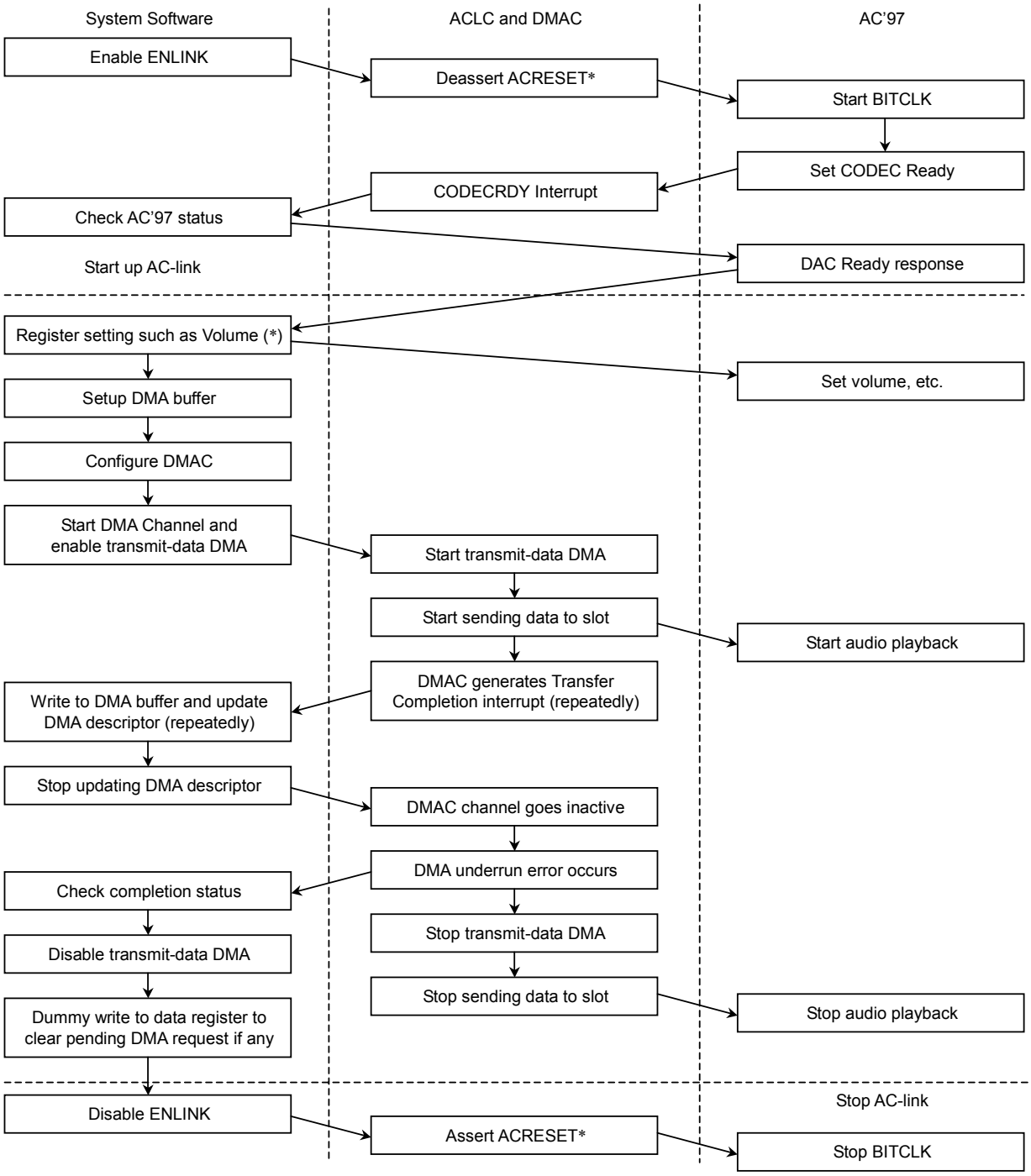
Figure 14.3.2 5.1 Channel Audio Connection Diagram

14.3.2 Boot Configuration

To utilize ACLC, the CPU must boot up with ACLC enabled by setting Pin Configuration Register's Shared Pin Select2 via the boot configuration. Refer to the sections 3.2 and 5.2.3 for the detail of the boot configuration.

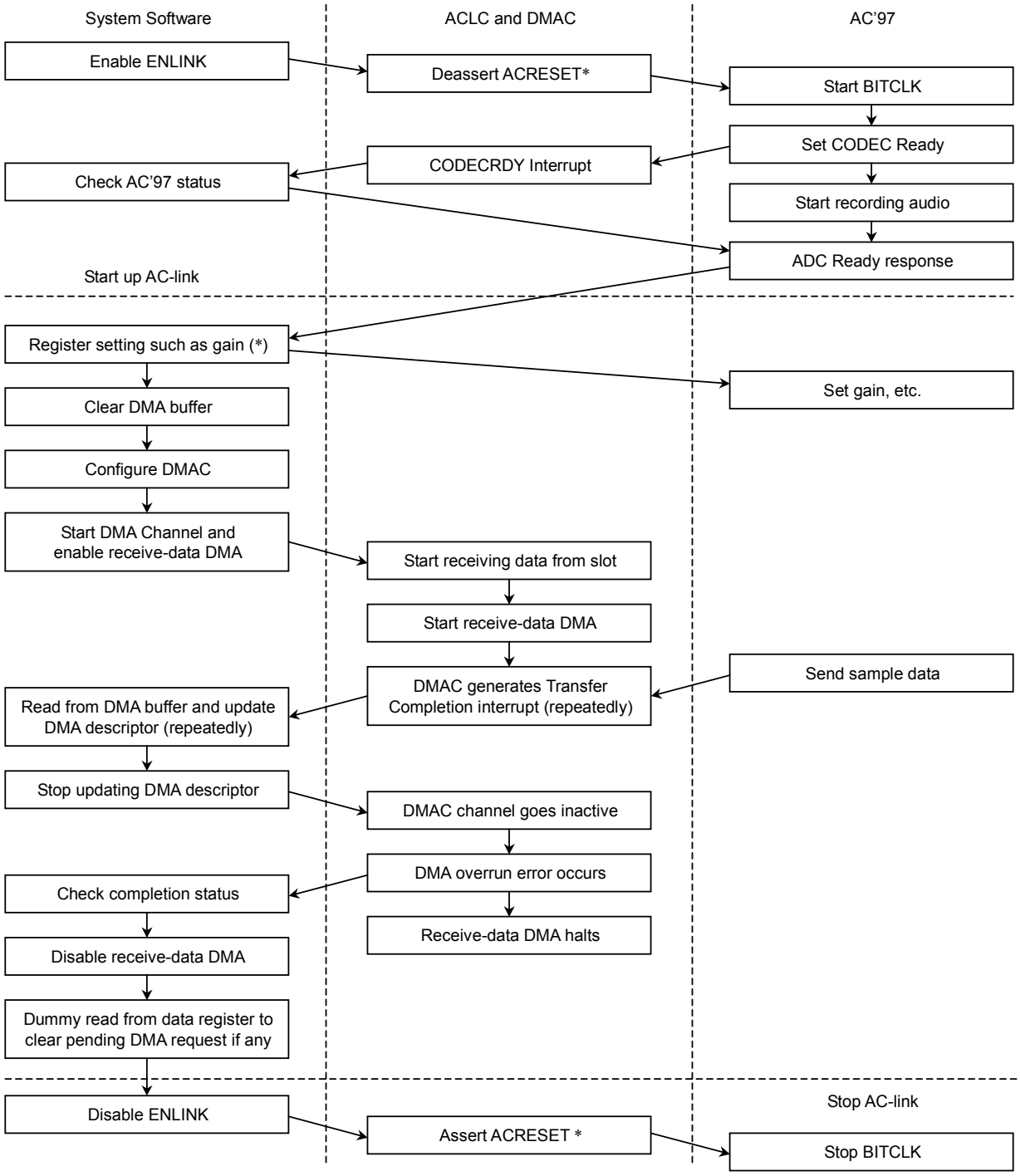
14.3.3 Usage Flow

This section outlines a process flow when using the AC'97 connected to ACLC. Refer to the subsequent sections for the details of each operation performed in this process flow. The diagrams below describe the audio playback and recording processes. The modem transmission and reception can be done in a similar way.



(*) Register settings such as volume can be made during data playback.

Figure 14.3.3 Audio Playback Process Flow



(*) Register settings such as gain can be made during data recording

Figure 14.3.4 Audio Recording Process Flow

14.3.4 AC-link Start Up

Figure 14.3.5 shows the conceptual sequence of AC-link start-up.

The ACLC Control Enable Register's Enable AC-link bit is used to deassert/assert the ACRESET* signal to the link side (including AC-link). This bit defaults to '0', so the CPU asserts the ACRESET* signal when it boots up.

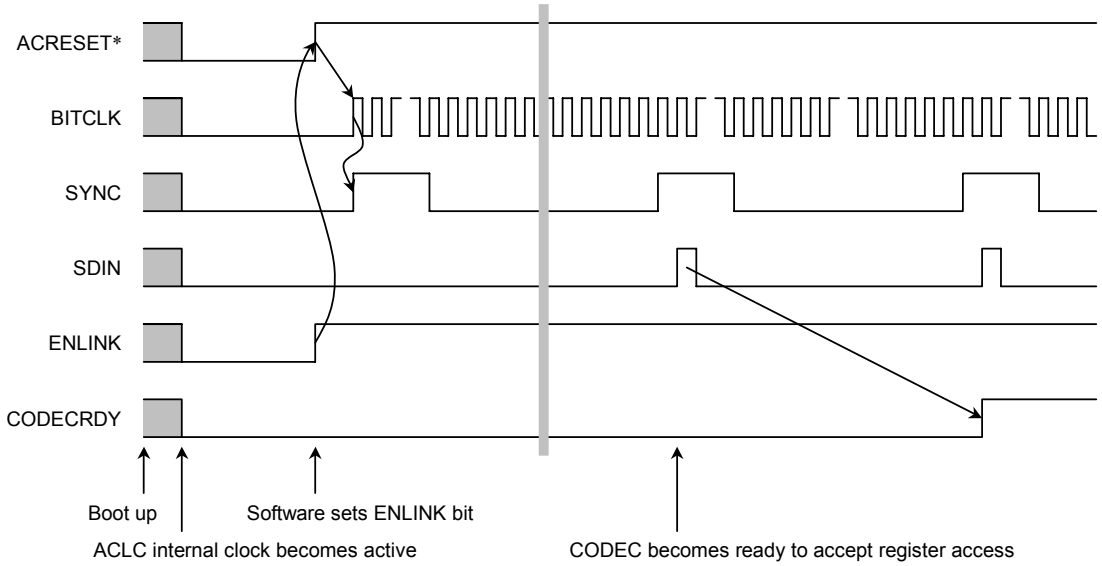
The AC'97 specification requires that the reset assertion period is 1µs or longer. The software is responsible for controlling the length of this period.

The AC'97 specification also requires that the primary CODEC stops the AC-link clock (BITCLK) signal during the period from ACRESET* signal assertion to 162.8ns after ACRESET* signal deassertion. ACLC assumes the primary CODEC meet this requirement.

Deasserting the link-side reset makes the primary CODEC start driving the BITCLK signal. When the BITCLK signal is provided, ACLC starts the SYNC signal output, which indicates the start of the AC-link frame, and starts the frame-length counting.

When a CODEC becomes ready to receive access to its own register, the CODEC sets the "CODEC Ready" bit of the Tag slot. When ACLC detects that this bit has been set, the ACLC Interrupt Status Register (ACINTSTS)'s CODEC[1:0] Ready (CODEC[1:0]RDY) bit is set. The system software is able to recognize the readiness of the CODEC(s) by detecting this event by way of either polling or interrupt.

In case of 5.1 channel audio connection example (Figure 14.3.2), because the secondary CODEC is connected to the SDIN1 signal of ACLC, the software must watch ACINTSTS.CODEC1RDY bit to determine the CODEC's readiness for the register access.



Note: The number of BITCLK cycles relative to other signals is not to scale.

Figure 14.3.5 Cold Reset and CODEC Ready Recognition

14.3.5 CODEC Register Access

By accessing registers in the CODEC, the system software is able to detect or control the CODEC state. This section describes how to read and write CODEC registers via ACLC. For details about AC'97 register set and proper sequence to operate CODEC, refer to the AC'97 specification and target CODEC datasheet.

It takes several frame periods for a read or write access to complete. Taking this into account, ACLC is equipped with a function for reporting CODEC register access completion as status-change or interrupt.

In order to read an AC'97 register, write the access destination CODEC ID and register address in ACLC CODEC Register Access Register (ACREGACC) with its CODECRD bit set to "1". After the ACLC Interrupt Status Register (ACINTSTS)'s REGACC Ready (REGACCRDY) bit is set, the software is able to get the data returned from the AC'97 by reading the ACREGACC register and issue another access.

In order to write to an AC'97 register, write the access destination CODEC ID, register address, and the data in ACLC's ACREGACC register with ACREGACC.CODECRD bit set to "0". After the ACINTSTS.REGACCRDY bit has been set, the software is able to issue another access.

In case of 5.1 channel audio connection example (Figure 14.3.2), because the secondary CODEC has CODEC ID of '3', the software must write '3' into ACREGACC.CODECID field when it issues secondary CODEC register access.

14.3.6 Sample-data Transmission and Reception

This section describes the mechanism for transmission and reception of PCM audio and modem wave-data. An overview is described first. The DMA (Direct Memory Access) operation, error detection and recovery procedure follow. A special case using slot activation control is described last.

14.3.6.1 Overview

Figure 14.3.6 and Figure 14.3.7 show conceptual views of the sample-data transmission and reception mechanisms.

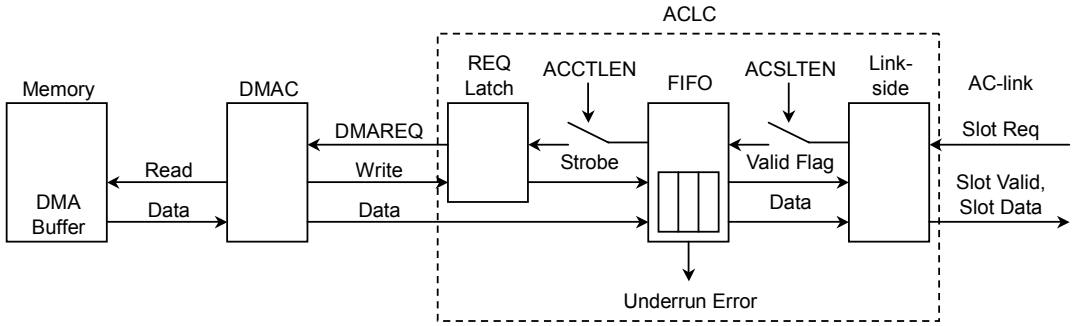


Figure 14.3.6 Sample-data Transmission Mechanism

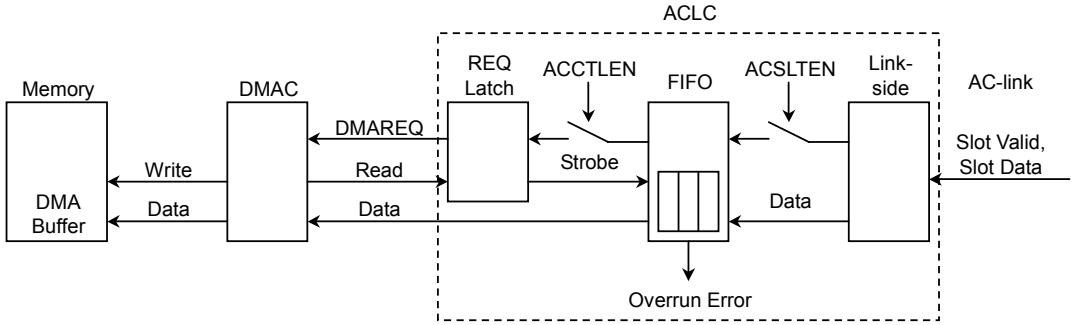


Figure 14.3.7 Sample-data Reception Mechanism

The CODEC requests ACLC to transmit and receive sample-data via ‘slot-request’ and ‘slot-valid’ bit-fields on the SDIN signal of AC-link.

For transmission, ACLC transmits the data with ‘slot-valid’ tag set. For reception, ACLC captures the slot-data.

Transmission or reception through each stream can be independently activated or deactivated under control of ACLC Slot Enable Register (ACSLTEN).

ACLC is equipped with a separate FIFO for each data-stream. The data to transmit is prefetched from memory to FIFO through DMA. The received data is buffered on FIFO and then stored to memory through DMA. In this stage, each DMA is independently activated or deactivated under control of ACLC Control Enable Register (ACCTLEN).

14.3.6.2 DMA Channel Mapping

ACLC uses four DMA request channels. These DMA channels are allocated to four out of seven data-streams, or slots, on the AC-link frame, according to ACLC DMA Channel Selection Register (ACDMASEL) setting as shown in Table 14.3.1. The pin configuration register allocates these DMA channels of ACLC to the DMAC (DMA controller) channels according to Pin Configuration Register (PCFG)'s DMA Request Selection (DMASEL[7:0]) bits as described in section 8.3.1.

Table 14.3.1 DMA Channel Mapping Modes

AC-link Slot Number	ACDMASEL			
	0	1	2	3
PCM L&R out (3&4)	ACLC ch0	ACLC ch0	ACLC ch0	ACLC ch0
Surround L&R out (7&8)		ACLC ch1	ACLC ch1	ACLC ch1
Center out (6)			ACLC ch2	
LFE out (9)			ACLC ch3	ACLC ch3
PCM L&R in (3&4) or Mic in (6)	ACLC ch1			ACLC ch2
Modem Line1 out (5)	ACLC ch2	ACLC ch2		
Modem Line 1 in (5)	ACLC ch3	ACLC ch3		

14.3.6.3 Sample-data Format

ACLC transmits/receives 16 bits per sample for each data slot shown in Table 14.3.1. The data resides on the first 16 bits of the 20 bits assigned to each slot on AC-link. Each sample-data register allows access by word (32-bit) unit only. Therefore the DMA count must be a multiple of word. Note that the transmit-data DMA count also must be the FIFO depth (refer to Table 14.3.8) or more for a reason described later.

For audio PCM front and surround streams, every data-word is loaded with a couple of left and right samples. For audio MIC stream, valid data is loaded in the same field as the left sample while the other field is filled with '0'. For audio center, LFE, and modem line 1 streams, two consecutive samples are packed into every word.

The data format at the sample-data register is arranged so that the data format on the DMA buffer follows the rules below.

- Each sample data is put in the byte order in which the CPU operates (big- or little-endian).
- Samples are put in the time-sequential order at increasing addresses on memory.
- For a DMA channel which couples left and right samples, each left sample precedes the corresponding right sample.

Refer to the sections 14.4.16 and later for the register format.

Figures below show the format of DMA buffer for each type of DMA channel. #0, #1, … means the sample's sequential number for the AC-link slot. Subscript 'L' means lower 8-bit of each sample and subscript '$_H$' means upper 8-bit.

Table 14.3.2 Front and Surround DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	Left#0_L	Left#0_H	Right#0_L	Right#0_H
+4	Left#1_L	Left#1_H	Right#1_L	Right#1_H
+8	Left#2_L	Left#2_H	Right#2_L	Right#2_H
:	:	:	:	:

Table 14.3.3 Center, LFE, and Modem DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_L	#0_H	#1_L	#1_H
+4	#2_L	#2_H	#3_L	#3_H
+8	#4_L	#4_H	#5_L	#5_H
:	:	:	:	:

Table 14.3.4 Mic DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_L	#0_H	0	0
+4	#1_L	#1_H	0	0
+8	#2_L	#2_H	0	0
:	:	:	:	:

Table 14.3.5 Front and Surround DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
+0	Left#0_H	Left#0_L	Right#0_H	Right#0_L
+4	Left#1_H	Left#1_L	Right#1_H	Right#1_L
+8	Left#2_H	Left#2_L	Right#2_H	Right#2_L
:	:	:	:	:

Table 14.3.6 Center, LFE, and Modem DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_H	#0_L	#1_H	#1_L
+4	#2_H	#2_L	#3_H	#3_L
+8	#4_H	#4_L	#5_H	#5_L
:	:	:	:	:

Table 14.3.7 Mic DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_H	#0_L	0	0
+4	#1_H	#1_L	0	0
+8	#2_H	#2_L	0	0
:	:	:	:	:

14.3.6.4 DMA Operation

When ACLC's REQ latch (refer to Figure 14.3.6 and Figure 14.3.7) needs to read or write sample-data, it issues a DMA request. When DMAC acknowledges the request by performing write- or read-access to the ACLC sample-data register, ACLC deasserts the request. Therefore, the software must properly set up DMAC so that the source or destination points to the corresponding sample-data register for the DMA channel.

Setup the DMA Channel Control Registers (DMCCRn) in DMAC as follows.

Immediate chain	Enable	DMCCRn.IMMCHN = 1 [Note]
DMA request polarity	Low-active	DMCCRn.REQPOL = 0
DMA acknowledge polarity	Low-active	DMCCRn.ACKPOL = 0
Request sense	Level-sensitive	DMCCRn.EGREQ = 0
Sample chain	1 word	DMCCRn.SMPCHN = 1
Transfer size	1 word	DMCCRn.XFSZ = 010b
Transfer address mode	Dual	DMCCRn.SNGAD = 0

Note: Use this setting when DMA chain operation is utilized

For a transmission channel, assign the address of ACLC Audio PCM Output/Surround/Center/LFE/Modem Output Register (ACAUDO/SURR/CENT/LFE/ MODODAT) to the DMAC destination address register (DMDARn). For a reception channel, assign the address of ACLC Audio input/Modem Input Register (ACAUDI/MODIDAT) to the DMAC source address register (DMSARn).

When any DMA request is pending, the REQ latch will not deasserted the request until the corresponding sample-data register is accessed. Just unsetting ACLC Control Enable Register (ACCTLEN)'s DMA Enable (xxxxDMA) bit corresponding to the DMA will not clear the REQ latch.

The procedure to continuously push or pull the sample-data stream through the chain DMA operation follows the DMAC specification. Refer to section 8.3.10 for this respect.

14.3.6.5 Sample-data FIFO

For a transmission stream, as long as ACLC Control Enable Register (ACCTLEN) allows that transmission and the FIFO has any room to fill data in, the FIFO issues a request via the REQ latch. On the other side, when a transmission FIFO receives a data-request from the link-side, it provides data with valid-flag set if it has any valid data. If it has no valid data, it responds with valid-flag unset and an underrun error bit is set.

At the transmit-data DMA start-up, until the FIFO becomes full, it responds to the link-side with valid-flag unset, in order to maximize the buffering effect. Therefore, the DMA size must be the FIFO depth or more.

Table 14.3.8 Transmission FIFO Depth

Data-stream	FIFO Depth (Word)
PCM L&R out	3
Surround L&R out	3
Center out	2
LFE out	2
Modem Line 1 out	1

The link-side drives the slot-valid bit and slot-data on AC-link. When underrun occurs, these bits are driven to all '0'.

For a reception stream, as long as the FIFO has any valid data, the FIFO issues a request via the REQ latch. On the other side, when ACCTLEN allows that reception and the link-side issues a data strobe, the FIFO stores the valid data. If the FIFO is full when it receives a data strobe, the data is discarded and an overrun error bit is set.

14.3.6.6 Error Detection and Recovery

In most usages, since the CODEC continuously requests sample-data transmission and reception, after DMA is finished, underrun and overrun will occur. The procedure described below allows the software to determine whether an error has occurred during DMA operation.

The software sets ACLC Control Enable Register (ACCTLEN)'s Error Halt Enable (xxxxEHLT) bit before it starts a DMA channel. After it starts the DMA channel, it waits until ACLC Interrupt Status Register (ACINTSTS)'s Underrun or Overrun Error (xxxxERR) bit is set. When the event is detected, the software checks DMA Channel Control Register (DMCCRn)'s Transfer Active (XFACT) bit and ACLC DMA Request Status Register (ACDMASTS)'s Request (xxxxDMA) bit and determines the DMA completion status as follows.

Table 14.3.9 DMA Completion Status Determination

DMCCRn.XFACT	ACDMASTS.xxxxDMA	Completion Status
Inactive	Pending	No Error during DMA
Inactive	Not Pending	Underrun or Overrun
Active	*	Underrun or Overrun

To recover from error, disable and enable the stream via ACCTLEN, and restart the DMA.

14.3.6.7 Slot Activation Control

In case ACLC is required to begin transmission or reception of multiple streams at the same time, slot activation control will be useful. To use this feature, the software must deactivate the relevant streams first, enable ACLC Control Enable Register (ACCTLEN), make sure the transmission FIFO becomes full by checking ACLC FIFO Status Register (ACFIFOSTS)'s Full (xxxxFULL) bit, and finally enable ACLC Slot Enable Register (ACSLTEN). This procedure assures that all the reception streams are activated at a frame and all the transmission streams begin to respond to the slot-request bits of that frame.

Note that access to ACSLTEN and ACLC Slot Disable Register (ACSLTDIS) needs special care to synchronize with the link-side. Refer to the register description for detail.

Since operating ACCTLEN register and DMAC without touching ACSLTEN is sufficient for most usages, the initial ACSLTEN value enables all the transmission and reception through the slots by default.

14.3.6.8 Variable Rate Limitation

To improve compatibility with existing AC'97 CODECs and controllers on the market, ACLC combines sample-data for the slots 3 and 4 into one DMA channel, and similarly for the slots 7 and 8. This feature effectively considers that the slot request bit from the CODEC for slot 4 shall be always same (in tandem) as for slot 3 for each frame, and similarly for the slots 7 and 8. ACLC also considers that the slot valid bit from the CODEC for slot 4 shall be always same (in tandem) as for slot 3 for each frame.

14.3.7 GPIO Operation

ACLC supports the slot 12 for the MC'97 (Modem Codec) GPIO.

The slot 12 is shadowed in the ACLC GPI Data Register (ACGPIDAT) and ACLC GPO Data Register (ACGPODAT) in the following way:

- ACLC copies the slot 12 input data into the ACGPIDAT register, if the slot 12 input is marked by the CODEC as valid in the AC-link frame period.
- ACLC generates the slot 12 output data from the ACGPODAT register and mark it as valid, if the slot 12 is required from the CODEC in the previous AC-link frame.

This shadowing function is enabled as long as ACSLTEN allows.

The bit 0 of the slot 12 is defined as 'GPIO_INT' and can cause ACLC to request an interrupt.

14.3.8 Interrupt

ACLC generate two kinds of interrupt to the interrupt controller as below.

- ACLC Interrupt

  Logical OR of all the valid bits of ACLC Interrupt Masked Status Register (ACINTMSTS) is connected. Refer to the section 14.4.5.

- ACLCPME Interrupt

  This interrupt shows the wake-up from CODEC in AC-link low-power mode.

  Refer to the description for ACLC Control Enable Register (ACCTLEN)'s Wake-up Enable (WAKEUP) bit in section 14.4.1.

14.3.9 AC-link Low-power Mode

The AC'97 specification makes provision for saving power during system suspension by powering-down both the controller and CODEC except the minimum circuit to detect modem RING/Caller-ID event and wake up the system. AC'97 CODEC is required to go into the low-power mode when they receive a special register-write access. In this mode, the AC-link controller must drive all output signals to low level to allow the CODEC digital I/O power cut.

ACLC provides 'AC-link low-power mode' setting. When this mode is enabled by ACLC Control Enable Register (ACCTLEN)'s Enable AC-link Low-power Mode (LOWPWR) bit, all the output signals except the ACRESET* signal to the AC-link are forced to low level.

The AC-link will be reactivated out of the low-power mode when the SYNC signal is driven high for 1 μs or longer by the AC-link controller while the BITCLK signal is inactive. The software is responsible for controlling the length of this period.

ACLC also provides the 'wake-up' function. While this function is enabled by ACCTLEN Register's Enable Wake-up (WAKEUP) bit, high-level input at any SDIN[x] signal will force ACLCPME interrupt assertion.

When ACLCPME interrupt is recognized, the software must disable the low-power mode and assert warm reset to the AC-link via ACCTLEN Register's Enable Warm Reset (WRESET) bit. After the warm reset is deasserted, the CODEC will start providing the BITCLK signal, and then ACLC will generate the SYNC signal for usual AC-link frames.

Refer to section B.5.1 of AC'97 specification revision 2.1 for the power-down and wake-up sequence in AC-link power-down mode.

14.4 Registers

The base address for the ACLC registers is described in section 4.2. Only word (32-bit) accesses are allowed. These registers return to their initial values when the module gets reset by power-on or configuration-register operation. The 'Disable AC-link' operation initializes the ACREGACC, ACGPIDAT, ACGPODAT, and ACSLTEN registers while keeping the other registers.

Do not access any location which is not mentioned in this section.

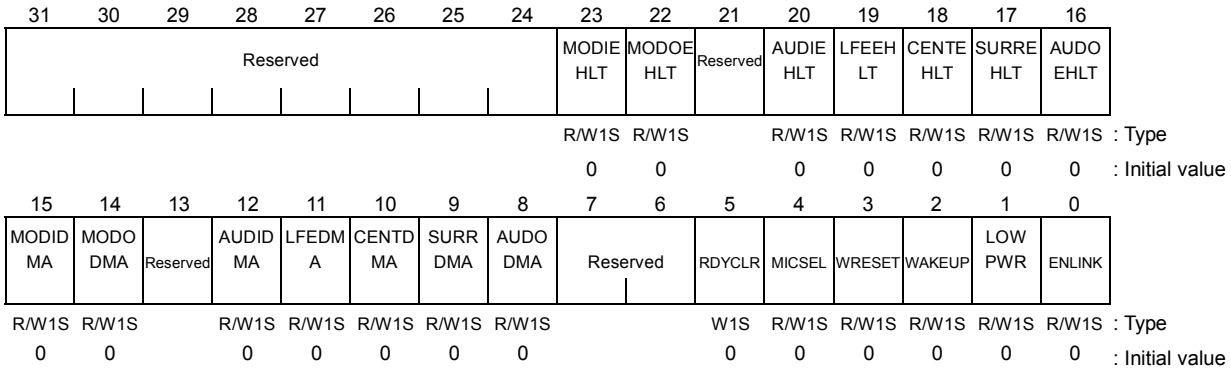
All the register bits marked as 'Reserved' are reserved. The value of the reserved bit when read is undefined. When any register is written, write to the reserved bit(s) the same value as the previous value read.

Table 14.4.1 ACLC Registers

Address	Mnemonic	Register Name	Type	Initial Value
0xF700	ACCTLEN	ACLC Control Enable Register	R/W1S	0x00000000
0xF704	ACCTLDIS	ACLC Control Disable Register	W1C	—
0xF708	ACREGACC	ACLC CODEC Register Access Register	R/W	0x00000000
0xF710	ACINTSTS	ACLC Interrupt Status Register	R/W1C	0x00000010
0xF714	ACINTMSTS	ACLC Interrupt Masked Status Register	R	0x00000000
0xF718	ACINTEN	ACLC Interrupt Enable Register	R/W1S	0x00000000
0xF71C	ACINTDIS	ACLC Interrupt Disable Register	W1C	—
0xF720	ACSEMAPH	ACLC Semaphore Register	RS/WC	0x00000000
0xF740	ACGPIDAT	ACLC GPI Data Register	R	0x00000000
0xF744	ACGPODAT	ACLC GPO Data Register	R/W	0x00000000
0xF748	ACSLTEN	ACLC Slot Enable Register	R/W1S	0x000003DF
0xF74C	ACSLTDIS	ACLC Slot Disable Register	W1C	—
0xF750	ACFIFOSTS	ACLC FIFO Status Register	R	0x00000000
0xF780	ACDMASTS	ACLC DMA Request Status Register	R	0x00000000
0xF784	ACDMASEL	ACLC DMA Channel Selection Register	R/W	0x00000000
0xF7A0	ACAUDODAT	ACLC Audio PCM Output Data Register	W	—
0xF7A4	ACSURRDAT	ACLC Surround Data Register	W	—
0xF7A8	ACCENTDAT	ACLC Center Data Register	W	—
0xF7AC	ACLFEDAT	ACLC LFE Data Register	W	—
0xF7B8	ACMODODAT	ACLC Modem Output Data Register	W	—
0xF7B0	ACAUDIDAT	ACLC Audio PCM Input Data Register	R	0xXXXXXXXX
0xF7BC	ACMODIDAT	ACLC Modem Input Data Register	R	0xXXXXXXXX
0xF7FC	ACREVID	ACLC Revision ID Register	R	0x00000203

14.4.1 ACLC Control Enable Register 0xF700

This register is used to check the setting of various ACLC features and to enable them.



Bit	Mnemonic	Field Name	Description	Read/Write
31:24	—	Reserved		—
23		Enable Modem Receive-data DMA Error Halt	MODIEHLT: Enable Modem Receive-data DMA Error Halt. R 0: Indicates that MODIDMA error halt is disabled. 1: Indicates that MODIDMA error halt is enabled. W1S 0: No effect 1: Enables MODIDMA error halt. When MODIDMA overrun occurs, subsequent DMA will not be issued.	R/W1S
22	MODOEHLT	Enable Modem Transmit-data DMA Error Halt	MODOEHLT: Enable Modem Transmit-data DMA Error Halt. R 0: Indicates that MODODMA error halt is disabled. 1: Indicates that MODODMA error halt is enabled. W1S 0: No effect 1: Enables MODODMA error halt. When MODODMA underrun occurs, subsequent DMA will not be issued.	R/W1S
21	—	Reserved		—
20	AUDIEHLT	Enable Audio Receive-data DMA Error Halt	AUDIEHLT: Enable Audio Receive-data DMA Error Halt. R 0: Indicates that AUDIDMA error halt is disabled. 1: Indicates that AUDIDMA error halt is enabled. W1S 0: No effect 1: Enables AUDIDMA error halt. When AUDIDMA overrun occurs, subsequent DMA request will not be issued.	R/W1S
19	LFEEHLT	Enable Audio LFE Transmit-data DMA Error Halt	LFEEHLT: Enable Audio LFE Transmit-data DMA Error Halt. R 0: Indicates that LFEDMA error halt is disabled. 1: Indicates that LFEDMA error halt is enabled. W1S 0: No effect 1: Enables LFEDMA error halt. When LFEDMA underrun occurs, subsequent DMA request will not be issued.	R/W1S
18	CENTEHLT	Enable Audio Center Transmit-data DMA Error Halt	CENTEHLT: Enable Audio Center Transmit-data DMA Error Halt. R 0: Indicates that CENTDMA error halt is disabled. 1: Indicates that CENTDMA error halt is enabled. W1S 0: No effect 1: Enables CENTDMA error halt. When CENTDMA underrun occurs, subsequent DMA request will not be issued.	R/W1S

Figure 14.4.1 ACCTLEN Register (1/3)

Bit	Mnemonic	Field Name	Description	Read/Write
17	SURREHLT	Enable Audio Surround L&R Transmit-data DMA Error Halt	SURREHLT: Enable Audio Surround L&R Transmit-data DMA Error Halt. R 0: Indicates that SURRDMA error halt is disabled. 1: Indicates that SURRDMA error halt is enabled. W1S 0: No effect 1: Enables SURRDMA error halt. When SURRDMA underrun occurs, subsequent DMA request will not be issued.	R/W1S
16	AUDOEHLT	Enable Audio PCM L&R Transmit-data DMA Error Halt	AUDOEHLT: Enable Audio PCM L&R Transmit-data DMA Error Halt. R 0: Indicates that AUDODMA error halt is disabled. 1: Indicates that AUDODMA error halt is enabled. W1S 0: No effect 1: Enables AUDODMA error halt. When AUDODMA underrun occurs, subsequent DMA request will not be issued.	R/W1S
15	MODIDMA	Enable Modem Receive-data DMA	MODIDMA: Enable Modem Receive-data DMA. R 0: Indicates that modem receive-data DMA is disabled. 1: Indicates that modem receive-data DMA is enabled. W1S 0: No effect 1: Enables modem receive-data DMA.	R/W1S
14	MODODMA	Enable Modem Transmit-data DMA	MODODMA: Enable Modem Transmit-data DMA. R 0: Indicates that modem transmit-data DMA is disabled. 1: Indicates that modem transmit-data DMA is enabled. W1S 0: No effect 1: Enables modem transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]	R/W1S
13	—	Reserved		—
12	AUDIDMA	Enable Audio Receive-data DMA	AUDIDMA: Enable Audio Receive-data DMA. R 0: Indicates that audio receive-data DMA is disabled. 1: Indicates that audio receive-data DMA is enabled. W1S 0: No effect 1: Enables audio receive-data DMA.	R/W1S
11	LFEDMA	Enable Audio LFE Transmit-data DMA	LFEDMA: Enable Audio LFE Transmit-data DMA. R 0: Indicates that audio LFE transmit-data DMA is disabled. 1: Indicates that audio LFE transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio LFE transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]	R/W1S
10	CENTDMA	Enable Audio Center Transmit- data DMA	CENTDMA: Enable Audio Center Transmit-data DMA. R 0: Indicates that audio Center transmit-data DMA is disabled. 1: Indicates that audio Center transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio Center transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]	R/W1S
9	SURRDMA	Enable Audio Surround L&R Transmit-data DMA	SURRDMA: Enable Audio Surround L&R Transmit-data DMA. R 0: Indicates that audio Surround L&R transmit-data DMA is disabled. 1: Indicates that audio Surround L&R transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio Surround L&R transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]	R/W1S
8	AUDODMA	Enable Audio PCM L&R Transmit-data DMA	AUDODMA: Enable Audio PCM L&R Transmit-data DMA. R 0: Indicates that audio PCM L&R transmit-data DMA is disabled. 1: Indicates that audio PCM L&R transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio PCM L&R transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]	R/W1S

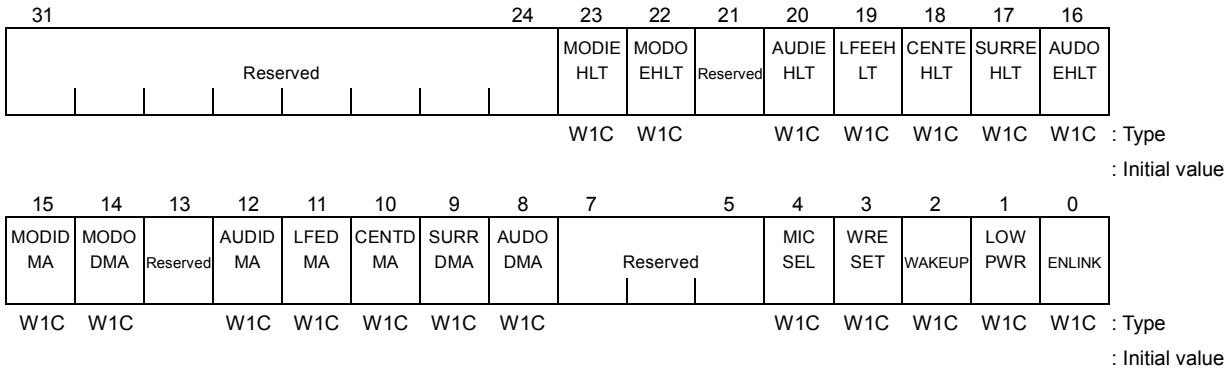
Figure 14.4.1 ACCTLEN Register (2/3)

Bit	Mnemonic	Field Name	Description	Read/Write
7:6	—	Reserved		—
5	RDYCLR	Clear CODEC Ready Bit	RDYCLR: Clear CODEC Ready Bit W1C 0: No effect 1: Clear CODEC[1:0] ready bits [Note: This bit should only be written to reevaluate the CODEC ready status after power-down command is sent to CODEC.]	W1S
4	MICSEL	MIC Selection	MICSEL: MIC Selection. R 0: Indicates that PCM L&R (Slot 3&4) is selected for audio reception. 1: Indicates that MIC (Slot 6) is selected for audio reception. W1S 0: No effect 1: Selects MIC (Slot 6) for audio reception.	R/W1S
3	WRESET	Assert Warm Reset	WRESET: Assert Warm Reset. R 0: Indicates that warm reset is not asserted. 1: Indicates that warm reset is asserted. W1S 0: No effect 1: Asserts warm reset. [Note 1: Do not assert warm reset during normal operation.] [Note 2: The software must guarantee the warm reset assertion time meets the AC'97 specification (1.0 μs or more).]	R/W1S
2	WAKEUP	Enable Wake-up	WAKEUP: Enable Wake-up. R 0: Indicates that wake-up from low-power mode is disabled. 1: Indicates that wake-up from low-power mode is enabled. While any SDIN signal is driven high, ACLC asserts ACLCPME interrupt request to the interrupt controller. W1S 0: No effect 1: Enables wake-up from low-power mode. [Note: Do not enable wake-up during normal operation.]	R/W1S
1	LOWPWR	Enable AC-link low-power mode	LOWPWR: Enable AC-link Low-power Mode. R 0: SYNC and SDOUT signals are not forced to low. 1: SYNC and SDOUT signals are forced to low. W1S 0: No effect 1: Forces SYNC and SDOUT signals low. [Note: Do not enable AC-link low-power mode during normal operation.]	R/W1S
0	ENLINK	Enable AC-link	ENLINK: Enable AC-link. R 0: Indicates that the ACRESET* signal to AC-link is asserted. 1: Indicates that the ACRESET* signal to AC-link is not asserted. W1S 0: No effect 1: Deasserts the ACRESET* signal to AC-link [Note: The software must guarantee the ACRESET* signal assertion time meets the AC'97 specification (1.0 μs or more).]	R/W1S

Figure 14.4.1 ACCTLEN Register (3/3)

14.4.2 ACLC Control Disable Register 0xF704

This register is used to disable various ACLC features.



Bit	Mnemonic	Field Name	Description	Read/Write
31:24	—	Reserved		—
23	MODIEHLT	Disable Modem Receive-data DMA Error Halt	MODIEHLT: Disable Modem Receive-data DMA Error Halt. W1C 0: No effect 1: Disables MODIDMA error halt. MODIDMA request(s) will continue to be issued even after MODIDMA overrun occurs.	W1C
22	MODOEHLT	Disable Modem Transmit-data DMA Error Halt	MODOEHLT: Disable Modem Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables MODODMA error halt. MODODMA request(s) will continue to be issued even after MODODMA underrun occurs.	W1C
21		Reserved		—
20	AUDIEHLT	Disable Audio Receive-data DMA Error Halt	AUDIEHLT: Disable Audio Receive-data DMA Error Halt. W1C 0: No effect 1: Disables AUDIDMA error halt. AUDIDMA request(s) will continue to be issued even after AUDIDMA overrun occurs.	W1C
19	LFEEHLT	Disable Audio LFE Transmit-data DMA Error Halt	LFEEHLT: Disable Audio LFE Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables LFEDMA error halt. LFEDMA request(s) will continue to be issued even after LFEDMA underrun occurs.	W1C
18	CENTEHLT	Disable Audio Center Transmit-data DMA Error Halt	CENTEHLT: Disable Audio Center Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables CENTDMA error halt. CENTDMA request(s) will continue to be issued even after CENTDMA underrun occurs.	W1C
17	SURREHLT	Disable Audio Surround L&R Transmit-data DMA Error Halt	SURREHLT: Disable Audio Surround L&R Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables SURRDMA error halt. SURRDMA request(s) will continue to be issued even after SURRDMA underrun occurs.	W1C
16	AUDOEHLT	Disable Audio PCM L&R Transmit-data DMA Error Halt	AUDOEHLT: Disable Audio PCM L&R Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables AUDODMA error halt. AUDODMA request(s) will continue to be issued even after AUDODMA underrun occurs.	W1C
15	MODIDMA	Disable Modem Receive-data DMA	MODIDMA: Disable Modem Receive-data DMA. W1C 0: No effect 1: Disables modem receive-data DMA.	W1C
14	MODODMA	Disable Modem Transmit-data DMA	MODODMA: Disable Modem Transmit-data DMA. W1C 0: No effect 1: Disables modem transmit-data DMA.	W1C

Figure 14.4.2 ACCTLDIS Register (1/2)

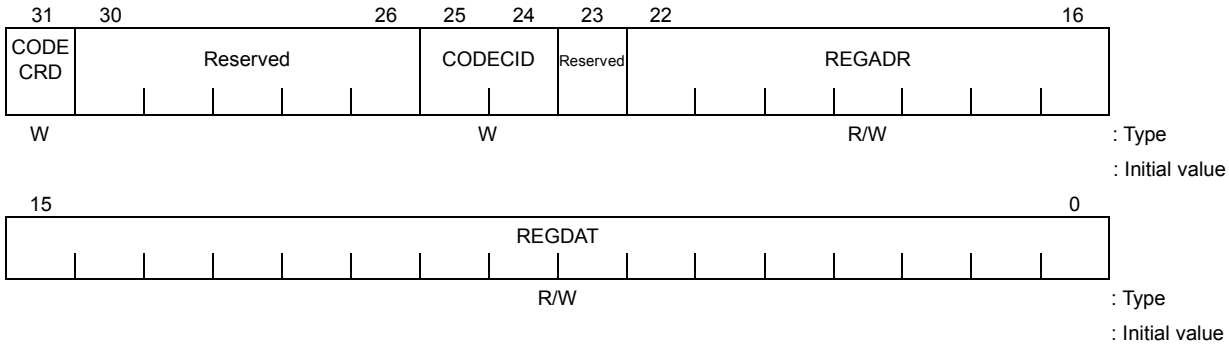
Bit	Mnemonic	Field Name	Description	Read/Write
13		Reserved		—
12		—	AUDIDMA: Disable Audio Receive-data DMA. W1C 0: No effect 1: Disables audio receive-data DMA.	W1C
11		—	LFEDMA: Disable Audio LFE Transmit-data DMA. W1C 0: No effect 1: Disables audio LFE transmit-data DMA.	W1C
10		—	CENTDMA: Disable Audio Center Transmit-data DMA. W1C 0: No effect 1: Disables audio Center transmit-data DMA.	W1C
9		—	SURRDMA: Disable Audio Surround L&R Transmit-data DMA. W1C 0: No effect 1: Disables audio Surround L&R transmit-data DMA.	W1C
8		—	AUDODMA: Disable Audio PCM L&R Transmit-data DMA. W1C 0: No effect 1: Disables audio PCM L&R transmit-data DMA.	W1C
7:5		Reserved		—
4		—	MICSEL: MIC Selection W1C 0: No effect 1: Selects PCM L&R (Slot 3&4) for audio reception	W1C
3		—	WRESET: Deassert Warm Reset. W1C 0: No effect 1: Deasserts warm reset. [Note: The software must guarantee the warm reset assertion time meets the AC'97 specification (1.0 μs or more).]	W1C
2		—	WAKEUP: Disable Wake-up. W1C 0: No effect 1: Disables wake-up from low-power mode.	W1C
1		—	LOWPWR: Disable AC-link Low-power Mode. W1C 0: No effect 1: Releases SYNC and SDOUT signals from low.	W1C
0		—	ENLINK: Disable AC-link. W1C 0: No effect 1: Asserts the ACRESET* signal to AC-link. [Note: The software must guarantee the ACRESET* signal assertion time meets the AC'97 specification (1.0 μs or more).]	W1C

Figure 14.4.2 ACCTLDIS Register (2/2)

Clear xxxxDMA bits in ACCTLEN to "0" by using this register to disable transmit/receive-data DMA and to stop transmission/reception by the AC-link. Note that if these bits are cleared while output-slot data is flowing in the FIFO, ACLC may output a wrong data as the last sample. This behavior will not occur if the software waits for data-flow completion by detecting underrun before it disables the corresponding slot.

14.4.3 ACLC CODEC Register Access Register 0xF708

CODEC registers can be accessed through this register.



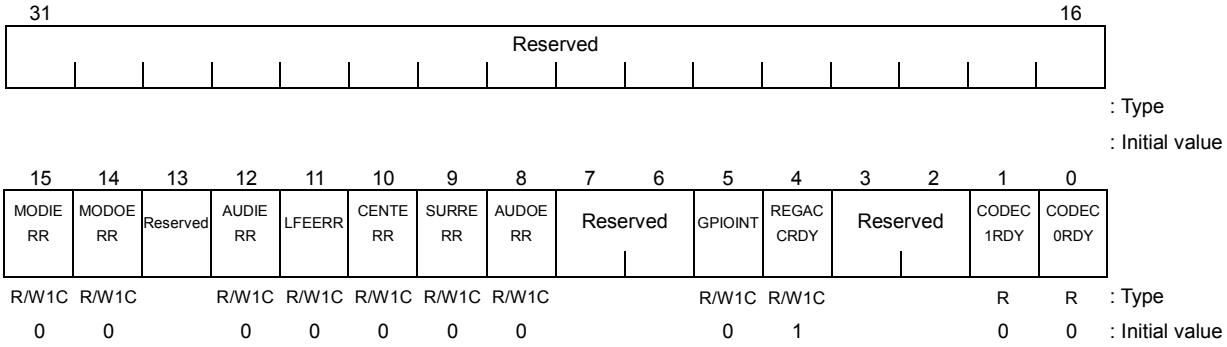
Bit	Mnemonic	Field Name	Description	Read/Write
31	CODECRD	AC'97 register read access	CODECRD: AC'97 register read access W 0: Indicates a write access. 1: Indicates a read access.	W
30:26	—	Reserved		—
25:24	CODECID	AC'97 CODEC ID	CODECID: AC'97 CODEC ID W Specifies the CODEC ID of the read/write access destination. The values "0" through "3" can be specified as the CODEC ID, but the number of CODECs actually supported depends on the configuration.	W
23	—	Reserved		—
22:16	REGADR	AC'97 register address	REGADR: AC'97 register address R Read address. Valid address can be read after read access is complete. W Specifies the read/write access destination address.	R/W
15:0	REGDAT	AC'97 register data	REGDAT: AC'97 register data R Read data. Valid data can be read after read access is complete. W Write data.	R/W

Figure 14.4.3 ACREGACC

This register must not be read from or written to until access completion is reported through the ACINTSTS register.

14.4.4 ACLC Interrupt Status Register 0xF710

This register shows various kinds of AC-link and ACLC status.



Bit	Mnemonic	Field Name	Description	Read/Write
31:16	—	Reserved		—
15	MODIERR	Modem Receive-data DMA Overrun	MODIERR: Modem Receive-data DMA Overrun R 1: Indicates that the modem receive-data DMA overran. W1C This bit is cleared when “1” is written to it.	R/W1C
14	MODOERR	Modem Transmit-data DMA Underrun	MODOERR: Modem Transmit-data DMA Underrun R 1: Indicates that the modem transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.	R/W1C
13	—	Reserved		—
12	AUDIERR	Audio Receive-data DMA Overrun	AUDIERR: Audio Receive-data DMA Overrun R 1: Indicates that the audio receive-data DMA overran. W1C This bit is cleared when “1” is written to it.	R/W1C
11	LFEERR	Audio LFE Transmit-data DMA Underrun	LFEERR: Audio LFE Transmit-data DMA Underrun R 1: Indicates that the audio LFE transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.	R/W1C
10	CENTERR	Audio Center Transmit-data DMA Underrun	CENTERR: Audio Center Transmit-data DMA Underrun R 1: Indicates that the audio center transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.	R/W1C
9	SURRERR	Audio Surround L&R Transmit-data DMA Underrun	SURRERR: Audio Surround L&R Transmit-data DMA Underrun R 1: Indicates that the audio surround L&R transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.	R/W1C
8	AUDOERR	Audio PCM L&R Transmit-data DMA Underrun	AUDOERR: Audio PCM L&R Transmit-data DMA Underrun R 1: Indicates that the audio PCM L&R transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.	R/W1C
7:6	—	Reserved		—
5	GPIOINT	GPIO Interrupt	GPIOINT: GPIO Interrupt R 1: Indicates that the incoming slot 12 bit[0] is ‘1’ (the modem CODEC GPIO interrupt). W1C This bit is cleared if “1” is written to it while the incoming slot 12 bit[0] is ‘0’.	R/W1C

Figure 14.4.4 ACINTSTS Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
4	REGACCRDY	ACREGACC Ready	REGACCRDY: ACREGACC Ready R 1: Indicates that the ACREGACC register is ready to get the value (in case the previous operation was a read access) and to initiate another R/W access to an AC'97 register. The result of reading or writing to the ACREGACC register before the completion notification is undefined. This bit is cleared if "1" is written to it. W1C This bit automatically becomes '0' when the ACREGACC register is written.	R/W1C
3:2	—	Reserved		—
1	CODEC1RDY	CODEC1 Ready	CODEC1RDY: CODEC1 Ready R 1: Indicates that the CODEC Ready bit of SDIN1 Slot0 is set.	R
0	CODEC0RDY	CODEC0 Ready	CODEC0RDY: CODEC0 Ready R 1: Indicates that the CODEC Ready bit of SDIN0 Slot0 is set.	R

Figure 14.4.4 ACINTSTS Register (2/2)

14.4.5 ACLC Interrupt Masked Status Register 0xF714

Every bit in this register is configured as follows:

ACINTMSTS = ACINTSTS & ACINTEN

Bit placement is the same as for the ACINTSTS register. The logical OR of all bits in this register is used as ACLC interrupt request to the interrupt controller.

14.4.6 ACLC Interrupt Enable Register 0xF718

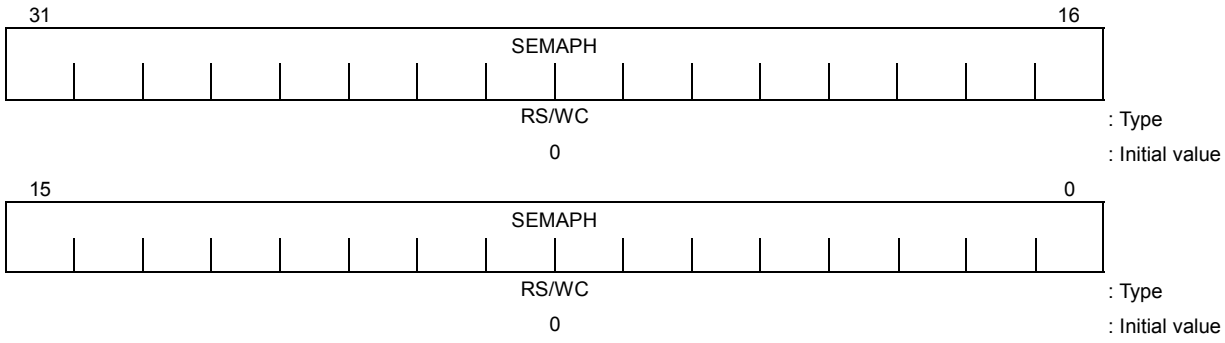
Interrupt request enable (R/W1S). Bit placement is the same as for the ACINTSTS register. Its initial value is all '0'. When a value is written to this register, the bit in the position where "1" was written is set to "1."

14.4.7 ACLC Interrupt Disable Register 0xF71C

Interrupt request enable clear (W1C). Bit placement is the same as for the ACINTSTS register. When a value is written to this register, the ACINTEN register bit in the position where a "1" was written is cleared to "0."

14.4.8 ACLC Semaphore Register 0xF720

This register is used for mutual exclusion control for resource.



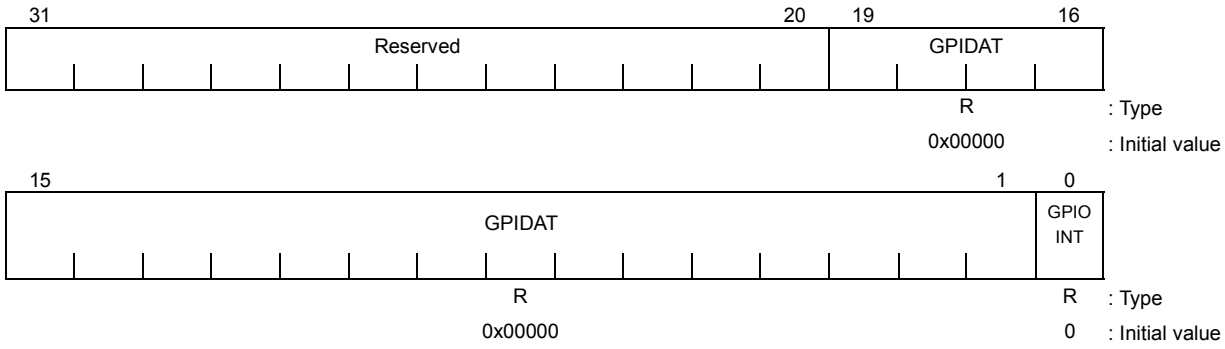
Bit	Mnemonic	Field Name	Description	Read/Write
31:0	SEMAPH	Semaphore flag	SEMAPH: Semaphore flag. RS 0: Indicates that the semaphore is unlocked. The read operation to this register will atomically set the bit[0] to lock the semaphore. 1: Indicates that the semaphore is locked. WC x: Writing any value to this register clears the bit[0] to release the semaphore.	RS/WC

Figure 14.4.5 ACSEMAPH Register

This register is provided primarily for the mutual exclusion between the audio and modem drivers to share the common resources of ACLC, such as the ACREGACC register and the link-control bits in the ACCTLEN/DIS register.

14.4.9 ACLC GPI Data Register 0xF740

This register shows GPIO (slot 12) input data.

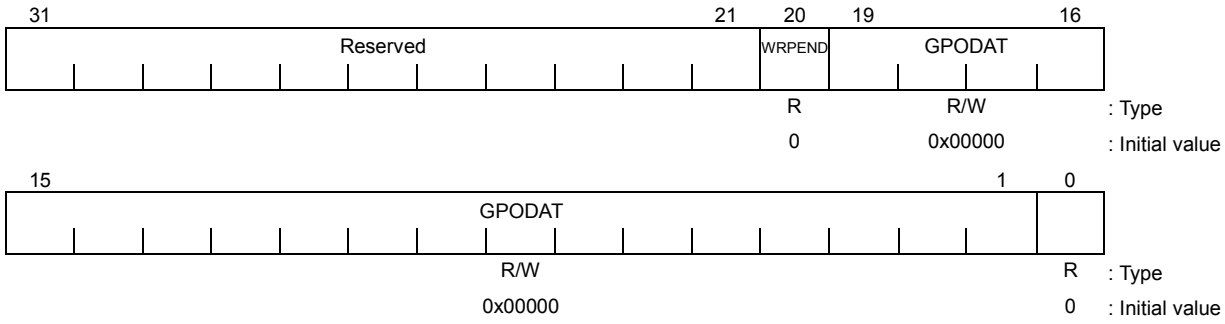


Bit	Mnemonic	Field Name	Description	Read/Write
31:20	—	Reserved		—
19:1	GPIDAT	GPIO-In data	GPIDAT: GPIO-In data R Read data. The incoming slot 12 bits[19:1] are shadowed here.	R
0	GPIOINT	GPIO Interrupt Indication	GPIOINT: GPIO Interrupt Indication R GPIO Interrupt. The incoming slot 12 bit[0] is shadowed here.	R

Figure 14.4.6 ACGPIDAT Register

14.4.10 ACLC GPO Data Register 0xF744

This register specifies GPIO (slot 12) output data.



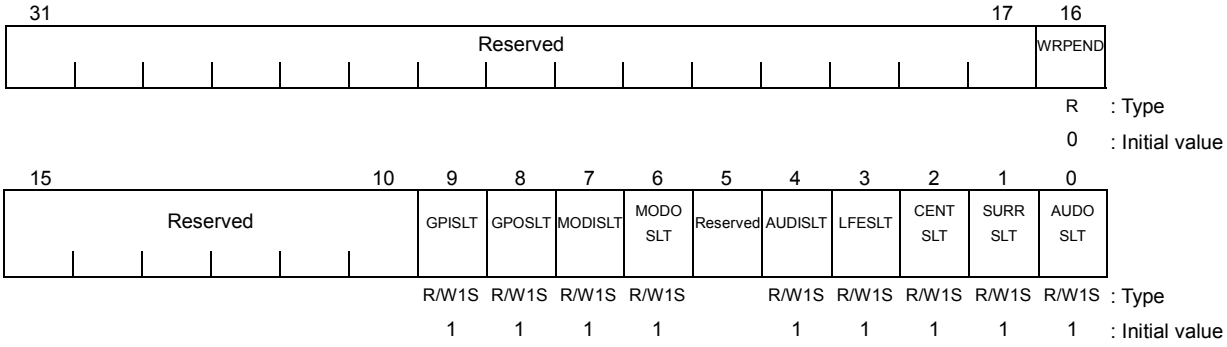
Bit	Mnemonic	Field Name	Description	Read/Write
31:20	—	Reserved		—
20	WRPEND	Write Pending	WRPEND: Write Pending R 0: Indicates that the previous write operation is complete and the ACGPODAT register is ready to be written. 1: Indicates that the previous write operation is not complete and the ACGPODAT register is not yet ready to be written.	R
19:1	GPODAT	GPIO-Out data	GPODAT: GPIO-Out data R Reads back the value previously written to this field. W Writes data to the outgoing slot 12 bits[19:1].	R/W
0	—	—	R Reads always '0'.	R

Figure 14.4.7 ACGPODAT Register

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACGPODAT.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

14.4.11 ACLC Slot Enable Register 0xF748

This register enables independently the AC-link slot data streams.



Bit	Mnemonic	Field Name	Description	Read/Write
31:17	—	Reserved		—
16	WRPEND	Write Pending	WRPEND: Write Pending R 0: Indicates that the previous write operation is complete and the ACSLTEN and ACSLTDIS registers are ready to be accessed. 1: Indicates that the previous write operation is not complete and the ACSLTEN and ACSLTEDIS registers are not yet ready to be accessed.	R
15:10	—	Reserved		—
9	GPISLT	Enable GPI slot reception	GPISLT: Enable GPI slot reception. R 0: Indicates that GPI slot reception is disabled. 1: Indicates that GPI slot reception is enabled. W1S 0: No effect 1: Enables GPI slot reception.	R/W1S
8	GPOSLT	Enable GPO Slot transmission	GPOSLT: Enable GPO Slot transmission. R 0: Indicates that GPO slot transmission is disabled. 1: Indicates that GPO slot transmission is enabled. W1S 0: No effect 1: Enables GPO slot transmission.	R/W1S
7	MODISLT	Enable Modem slot reception	MODISLT: Enable Modem slot reception. R 0: Indicates that modem slot reception is disabled. 1: Indicates that modem slot reception is enabled. W1S 0: No effect 1: Enables modem slot reception.	R/W1S
6	MODOSLT	Enable Modem slot transmission	MODOSLT: Enable Modem slot transmission. R 0: Indicates that modem slot transmission is disabled. 1: Indicates that modem slot transmission is enabled. W1S 0: No effect 1: Enables modem slot transmission.	R/W1S
5	—	Reserved		—

Figure 14.4.8 ACSLTEN Register (1/2)

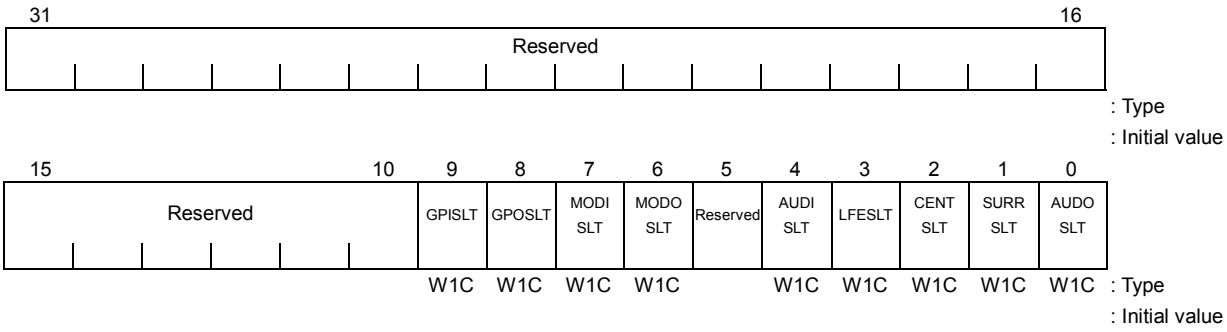
Bit	Mnemonic	Field Name	Description	Read/Write
4	AUDISLT	Enable Audio slot reception	AUDISLT: Enable Audio slot reception. R 0: Indicates that audio slot reception is disabled. 1: Indicates that audio slot reception is enabled. W1S 0: No effect 1: Enables audio slot reception.	R/W1S
3	LFESLT	Enable Audio LFE slot transmission	LFESLT: Enable Audio LFE slot transmission. R 0: Indicates that audio LFE slot transmission is disabled. 1: Indicates that audio LFE slot transmission is enabled. W1S 0: No effect 1: Enables audio LFE slot transmission.	R/W1S
2	CENTSLT	Enable Audio Center slot transmission	CENTSLT: Enable Audio Center slot transmission. R 0: Indicates that audio Center slot transmission is disabled. 1: Indicates that audio Center slot transmission is enabled. W1S 0: No effect 1: Enables audio Center slot transmission.	R/W1S
1	SURRSLT	Enable Audio Surround L&R slot transmission	SURRSLT: Enable Audio Surround L&R slot transmission. R 0: Indicates that audio Surround L&R slot transmission is disabled. 1: Indicates that audio Surround L&R slot transmission is enabled. W1S 0: No effect 1: Enables audio Surround L&R slot transmission.	R/W1S
0	AUDOSLT	Enable Audio PCM L&R slot transmission	AUDOSLT: Enable Audio PCM L&R slot transmission. R 0: Indicates that audio PCM L&R Slot transmission is disabled. 1: Indicates that audio PCM L&R Slot transmission is enabled. W1S 0: No effect 1: Enables audio PCM L&R slot transmission.	R/W1S

Figure 14.4.8 ACSLTEN Register (2/2)

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACSLTEN.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

14.4.12 ACLC Slot Disable Register 0xF74C

This register disables independently the AC-link slot data streams.



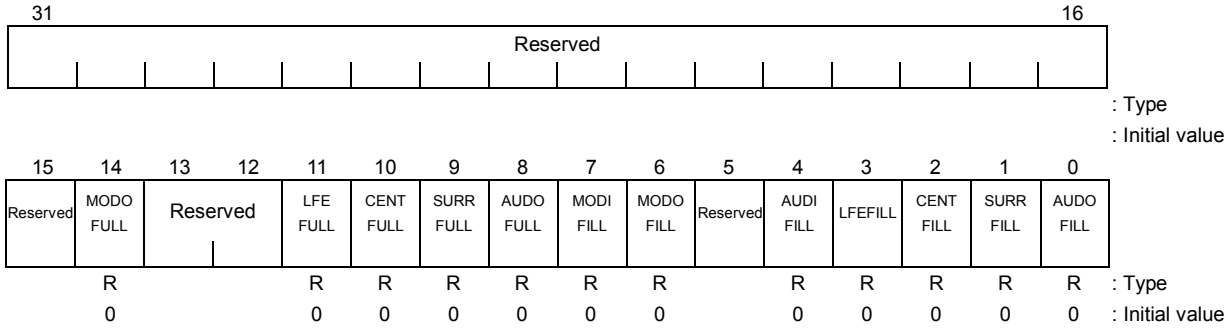
Bit	Mnemonic	Field Name	Description	Read/Write
31:10	—	Reserved		—
9	GPISLT	Disable GPI slot reception	GPISLT: Disable GPI slot reception. W1C 0: No effect 1: Disables GPI slot reception.	W1C
8	GPOSLT	Disable GPO Slot transmission	GPOSLT: Disable GPO Slot transmission. W1C 0: No effect 1: Disables GPO slot transmission.	W1C
7	MODISLT	Disable Modem slot reception	MODISLT: Disable Modem slot reception. W1C 0: No effect 1: Disables modem slot reception.	W1C
6	MODOSLT	Disable Modem slot transmission	MODOSLT: Disable Modem slot transmission. W1C 0: No effect 1: Disables modem slot transmission.	W1C
5	—	Reserved		—
4	AUDISLT	Disable Audio slot reception	AUDISLT: Disable Audio slot reception. W1C 0: No effect 1: Disables audio slot reception.	W1C
3	LFESLT	Disable Audio LFE slot transmission	LFESLT: Disable Audio LFE slot transmission. W1C 0: No effect 1: Disables audio LFE slot transmission.	W1C
2	CENTSLT	Disable Audio Center slot transmission	CENTSLT: Disable Audio Center slot transmission. W1C 0: No effect 1: Disables audio Center slot transmission.	W1C
1	SURRSLT	Disable Audio Surround L&R slot transmission	SURRSLT: Disable Audio Surround L&R slot transmission. W1C 0: No effect 1: Disables audio Surround L&R slot transmission.	W1C
0	AUDOSLT	Disable Audio PCM L&R slot transmission	AUDOSLT: Disable Audio PCM L&R slot transmission. W1C 0: No effect 1: Disables audio PCM L&R slot transmission.	W1C

Figure 14.4.9 ACSLTDIS Register

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACSLTEN.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

14.4.13 ACLC FIFO Status Register 0xF750

This register indicates the AC-link slot data FIFO status.



Bit	Mnemonic	Field Name	Description	Read/Write
31:15	—	Reserved		—
14	MODOFULL	Modem Transmit-data Full	MODOFULL: Modem Transmit-data Full. R 0: Indicates modem transmit-data FIFO is not full. 1: Indicates modem transmit-data FIFO is full.	R
13:12	Reserved	Reserved		—
11	LFEFULL	Audio LFE Transmit-data Full	LFEFULL: Audio LFE Transmit-data Full. R 0: Indicates audio LFE transmit-data FIFO is not full. 1: Indicates audio LFE transmit-data FIFO is full.	R
10	CENTFULL	Audio Center Transmit-data Full	CENTFULL: Audio Center Transmit-data Full. R 0: Indicates audio Center transmit-data FIFO is not full 1: Indicates audio Center transmit-data FIFO is full.	R
9	SURRFULL	Audio Surround L&R Transmit-data Full	SURRFULL: Audio Surround L&R Transmit-data Full. R 0: Indicates audio Surround L&R transmit-data FIFO is not full. 1: Indicates audio Surround L&R transmit-data FIFO is full.	R
8	AUDOFULL	Audio PCM L&R Transmit-data Full	AUDOFULL: Audio PCM L&R Transmit-data Full. R 0: Indicates audio PCM L&R transmit-data FIFO is not full. 1: Indicates audio PCM L&R transmit-data FIFO is full.	R
7	MODIFILL	Modem Receive-data Filled	MODIFILL: Modem Receive-data Filled. R 0: Indicates modem receive-data FIFO is empty. 1: Indicates modem receive-data FIFO is not empty.	R
6	MODOFILL	Modem Transmit-data Filled	MODOFILL: Modem Transmit-data Filled. R 0: Indicates modem transmit-data FIFO is empty. 1: Indicates modem transmit-data FIFO is not empty.	R
5	—	Reserved		—
4	AUDIFILL	Audio Receive-data Filled	AUDIFILL: Audio Receive-data Filled. R 0: Indicates audio receive-data FIFO is empty. 1: Indicates audio receive-data FIFO is not empty.	R
3	LFEFILL	Audio LFE Transmit-data Filled	LFEFILL: Audio LFE Transmit-data Filled. R 0: Indicates audio LFE transmit-data FIFO is empty. 1: Indicates audio LFE transmit-data FIFO is not empty.	R

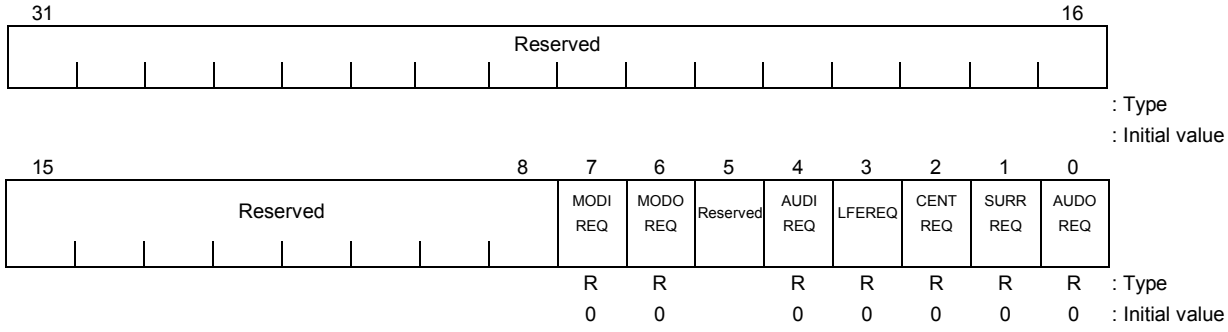
Figure 14.4.10 ACFIFOSTS Register (1/2)

Bit	Mnemonic	Field Name	Description	Read/Write
2	CENTFILL	Audio Center Transmit-data Filled	CENTFILL: Audio Center Transmit-data Filled. R 0: Indicates audio Center transmit-data FIFO is empty. 1: Indicates audio Center transmit-data FIFO is not empty.	R
1	SURRFILL	Audio Surround L&R Transmit-data Filled	SURRFILL: Audio Surround L&R Transmit-data Filled. R 0: Indicates audio Surround L&R transmit-data FIFO is empty. 1: Indicates audio Surround L&R transmit-data FIFO is not empty.	R
0	AUDOFILL	Audio PCM L&R Transmit-data Filled	AUDOFILL: Audio PCM L&R Transmit-data Filled. R 0: Indicates audio PCM L&R transmit-data FIFO is empty. 1: Indicates audio PCM L&R transmit-data FIFO is not empty	R

Figure 14.4.10 ACFIFOSTS Register (2/2)

14.4.14 ACLC DMA Request Status Register 0xF780

This register indicates the AC-link slot data DMA request status.



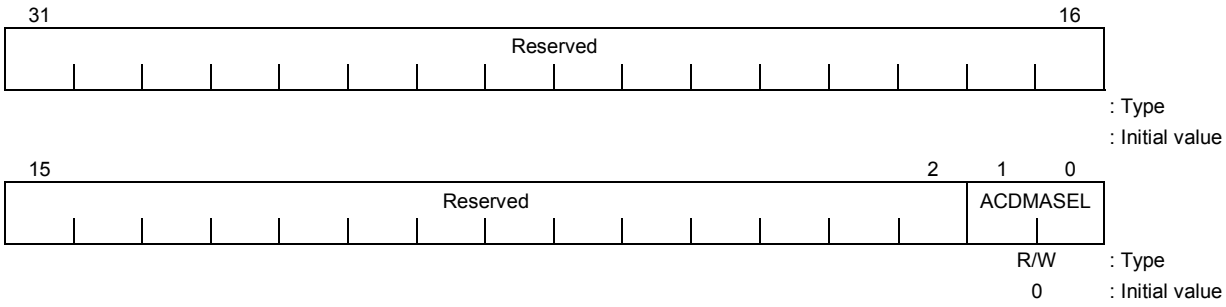
Bit	Mnemonic	Field Name	Description	Read/Write
31:8	—	Reserved		—
7	MODIREQ	Modem Data Reception Request	MODIREQ: Modem Data Reception Request R 0: No request is pending. 1: Request is pending.	R
6	MODOREQ	Modem Data Transmission Request	MODOREQ: Modem Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
5	—	Reserved		—
4	AUDIREQ	Audio Data Reception Request	AUDIREQ: Audio Data Reception Request R 0: No request is pending. 1: Request is pending.	R
3	LFEREQ	Audio LFE Data Transmission Request	LFEREQ: Audio LFE Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
2	CENTREQ	Audio Center Data Transmission Request	CENTREQ: Audio Center Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
1	SURRREQ	Audio Surround L&R Data Transmission Request	SURRREQ: Audio Surround L&R Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
0	AUDOREQ	Audio PCM L&R Data Transmission Request	AUDOREQ: Audio PCM L&R Data Transmission Request R 0: No request is pending. 1: Request is pending.	R

Figure 14.4.11 ACDMASTS Register

This read-only register shows if any DMA request is pending for each data I/O channel. A DMA request can be pending after the software deactivates the DMAC channel or disables DMA by ACCTLDIS register bit to complete DMA operation. In this case, write or read the sample data register (ACAUDODAT and others) to clear the DMA request.

14.4.15 ACLC DMA Channel Selection Register 0xF784

This register is used to select and check the channel allocation for AC-link slot data DMA.



Bit	Mnemonic	Field Name	Description	Read/Write
31:2	—	Reserved		—
1:0	ACDMASEL	DMA Channel Selection	ACDMASEL: DMA Channel Selection R/W ACDMASEL: DMA Channel Selection 0: PCM L&R out, Audio in, and Modem out&in. 1: PCM L&R out, Surround L&R out, and Modem out&in. 2: PCM L&R out, Surround L&R out, Center out, and LFE out. 3: PCM L&R out, Surround L&R out, Center out, and Audio in.	R/W

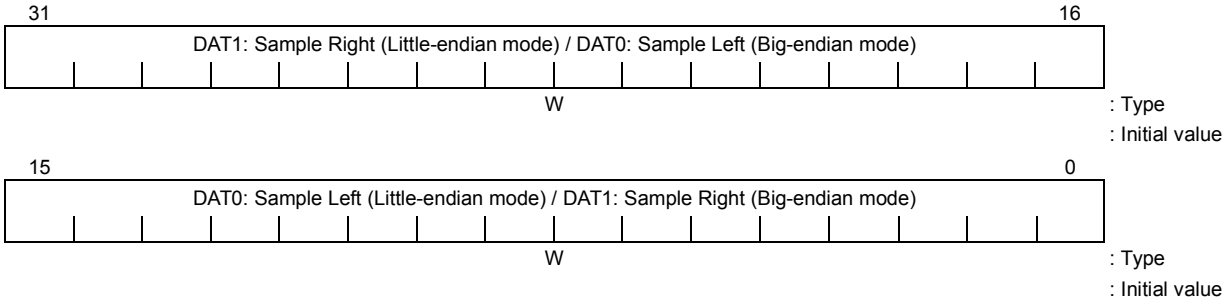
Figure 14.4.12 ACDMASEL Register

This register selects DMA channel mapping mode. The software is recommended to make sure no DMA request is pending before changing this register value.

14.4.16 ACLC Audio PCM Output Data Register 0xF7A0

ACLC Surround Data Register 0xF7A4

These registers are used to write audio PCM and surround L&R output data.



Bit	Mnemonic	Field Name		Description		Read/Write
				Little-endian mode	Big-endian mode	
31:16	—	—	W	DAT1: Sample Right	DAT0: Sample Left	W
15:0	—	—	W	DAT0: Sample Left	Left DAT1: Sample Right	W

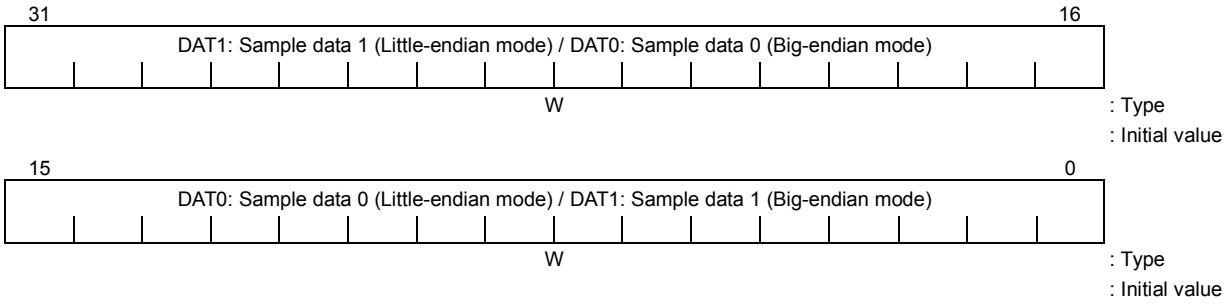
Figure 14.4.13 ACAUDODAT/ACSURRDAT Register

14.4.17 ACLC Center Data Register 0xF7A8

ACLC LFE Data Register 0xF7AC

ACLC Modem Output Data Register 0xF7B8

This registers are used to write audio center, LFE, and modem output data.

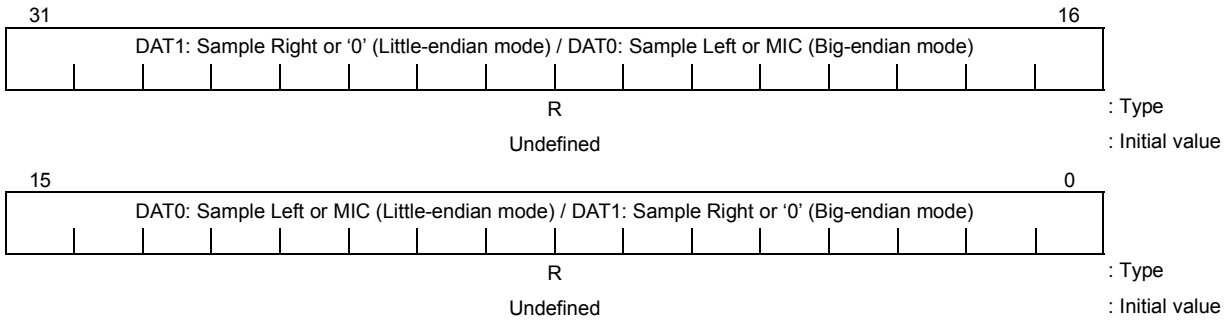


Bit	Mnemonic	Field Name		Description		Read/Write
				Little-endian mode	Big-endian mode	
31:16	—	—	W	DAT1: Sample data 1	DAT0: Sample data 0	W
15:0	—	—	W	DAT0: Sample data 0	DAT1: Sample data 1	W

Figure 14.4.14 ACCENDAT/ACLFEDAT/ACMODODAT Register

14.4.18 ACLC Audio PCM Input Data Register 0xF7B0

This register is used to read audio PCM input data.

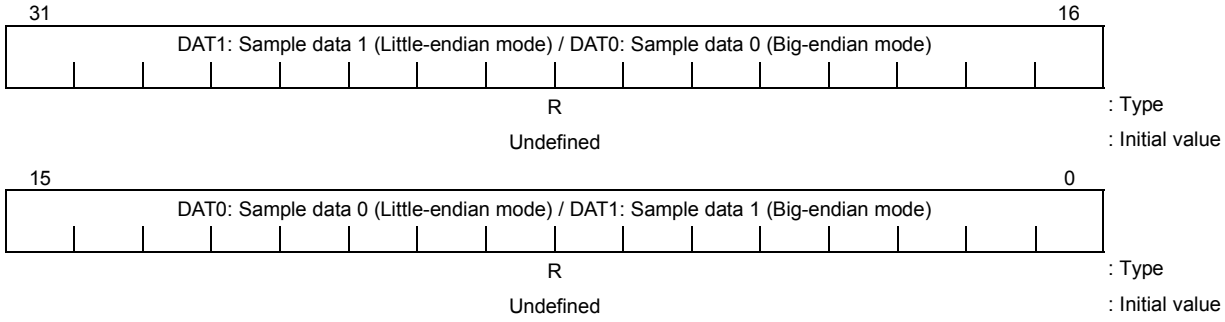


Bit	Mnemonic	Field Name		Description		Read/Write
				Little-endian mode	Big-endian mode	
31:16	—	—	R	DAT1: Sample Right or '0'	DAT0: Sample Left or MIC	R
15:0	—	—	R	DAT0: Sample Left or MIC	DAT1: Sample Right or '0'	R

Figure 14.4.15 ACAUDIDAT Register

14.4.19 ACLC Modem Input Data Register 0xF7BC

This register is used to read modem input data.

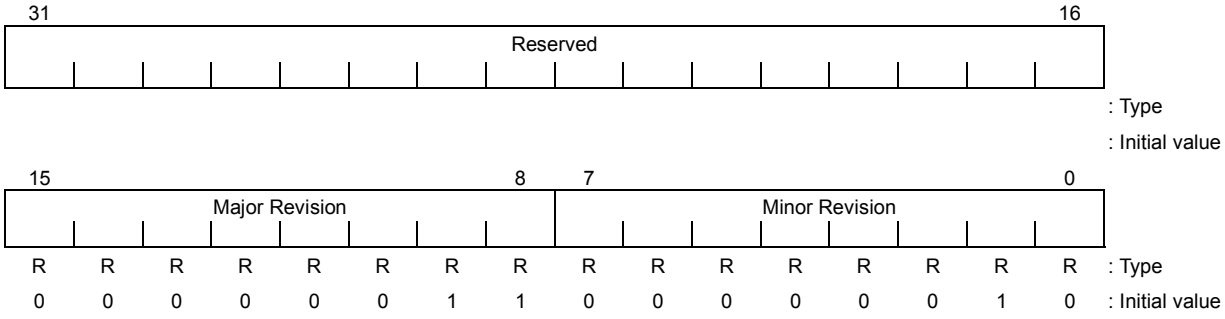


Bit	Mnemonic	Field Name		Description		Read/Write
				Little-endian mode	Big-endian mode	
31:16	—	—	R	DAT1: Sample data 1	DAT0: Sample data 0	R
15:0	—	—	R	DAT0: Sample data 0	DAT1: Sample data 1	R

Figure 14.4.16 ACMODIDAT Register

14.4.20 ACLC Revision ID Register 0xF7FC

This register is used to read ACLC module's revision ID.


31
16
Reserved
: Type
: Initial value
15
8
7
0
Major Revision
Minor Revision
R R R R R R R R R R R R R R R R : Type
0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 : Initial value

Bit	Mnemonic	Field Name	Description	Read/Write
31:16	—	Reserved		—
15:8	—	—	R Major Revision Contact Toshiba technical staff for an explanation of the revision value.	R
7:0	—	—	R Minor Revision Contact Toshiba technical staff for an explanation of the revision value.	R

Figure 14.4.17 ACREVID Register

This read-only register shows the revision of ACLC module. Note that this number is not related to the AC'97 specification revision.

15. Interrupt Controller

15.1 Characteristics

The TX4938 on-chip Interrupt Controller (IRC) receives interrupt requests from the TX4938 on-chip peripheral circuitry as well as external interrupt requests then generates interrupt requests to the TX49/H3 processor core.

Also, the Interrupt Controller has a 16-bit flag register that generates interrupt requests to either external devices or to the TX49/H3 core.

The Interrupt Controller has the following characteristics.

- Supports interrupts from 18 types of on-chip peripheral circuits and a maximum of 6 external interrupt signal inputs
- Sets 8 priority interrupt levels for each interrupt input
- Can select either edge detection or level detection for each external interrupt when in the interrupt detection mode
- As a flag register used for interrupt requests, the Interrupt Controller contains a 16-bit readable/writeable register and can issue interrupt requests to external devices as well as to the TX49/H3 core (IRC interrupt).

15.2 Block Diagram

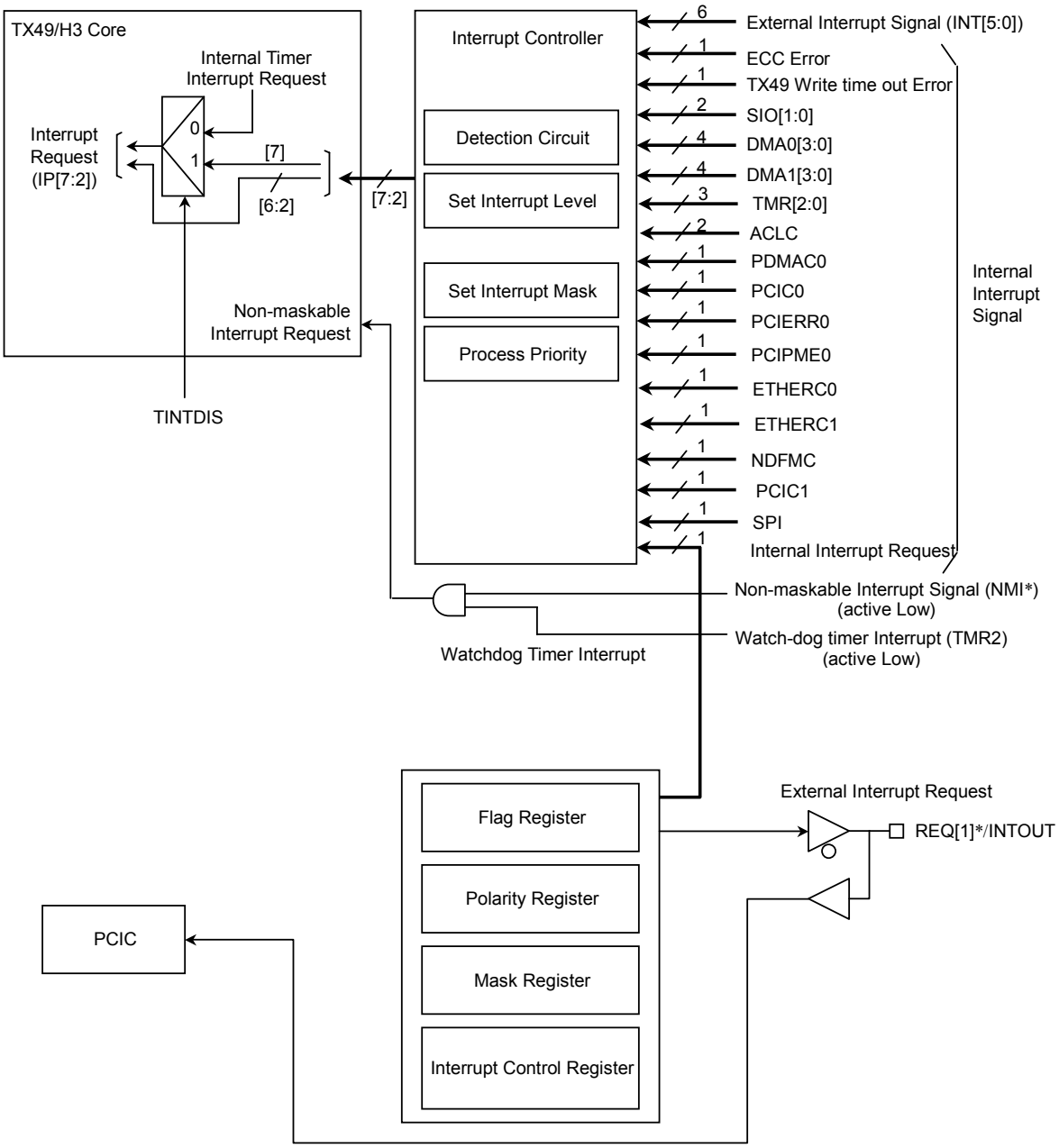


Figure 15.2.1 Interrupt Controller Outline

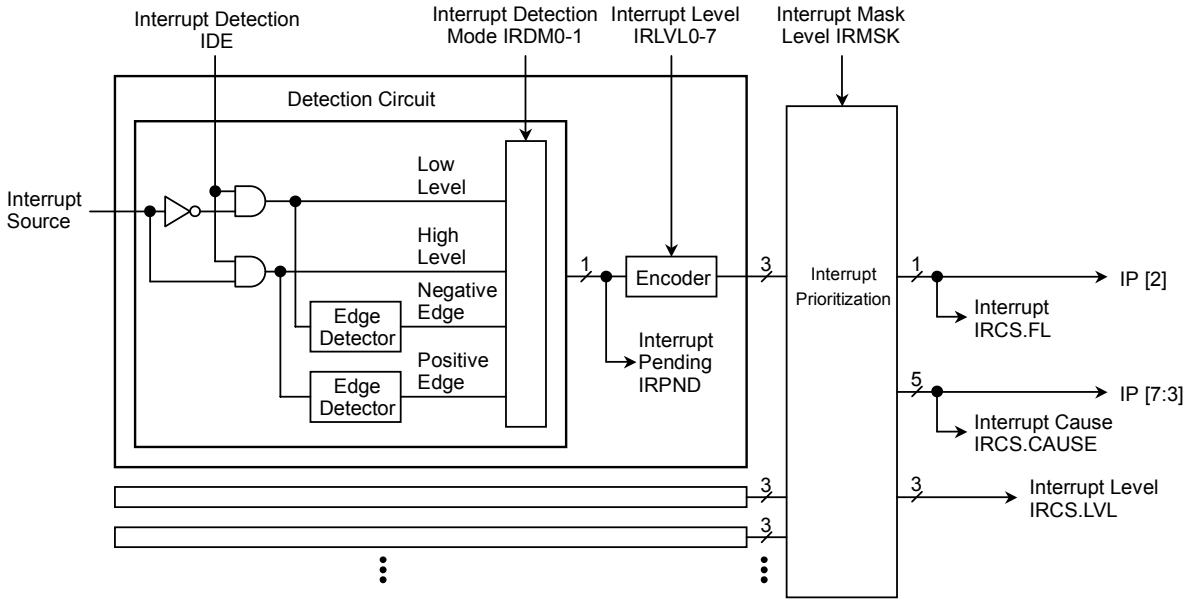


Figure 15.2.2 Internal Block Diagram of Interrupt Controller

15.3 Detailed Explanation

15.3.1 Interrupt sources

The TX4938 has as interrupt sources interrupts from 18 types of on-chip peripheral circuits and 6 external interrupt signals.

Table 15.3.1 lists the interrupt sources. Signals with the lower interrupt number have the higher priority. The priorities are explained below in section 15.3.4.

The interrupt number 5 is shared between INT[3] and ETHERC1. When ETHERC1 is used, an interrupt is noticed from ETHERC1 to the IRC interrupt number 5 internal of a chip. The external pin INT[3] is used when ETHERC1 is used for the purpose other than an interrupt. This is the same as for the interrupt number 6. Refer to "3.3 Pin multiplex" for muliplexed pins.

Table 15.3.1 Interrupt Sources and Priorities

Priority	Interrupt Number	Interrupt Source
High	0	SDRAM ECC Error (Internal)
	1	TX49 Write Timeout Error (Internal)
	2	INT[0] (External)
	3	INT[1] (External)
	4	INT[2] (External)
	5	INT[3] (External) / ETHERC1 (Internal)
	6	INT[4] (External) / ETHERC0 (Internal)
	7	INT[5] (External)
	8	SIO0 (Internal)
	9	SIO1 (Internal)
	10	DMA0[0] (Internal)
	11	DMA0[1] (Internal)
	12	DMA0[2] (Internal)
	13	DMA0[3] (Internal)
	14	IRC (Internal)
	15	PDMAC0 (Internal)
	16	PCIC0 (Internal)
	17	TMR0 (Internal)
	18	TMR1 (Internal)
	19	TMR2 (Internal)
	20	(Reserved)
	21	NDFMC (Internal)
	22	PCIERR (Internal)
	23	PCIPMC (Internal)
	24	ACLC (Internal)
	25	ACLCPME (Internal)
	26	PCIC1INT (Internal)
	27	DMAC1[0] (Internal)
	28	DMAC1[1] (Internal)
	29	DMAC1[2] (Internal)
	30	DMAC1[3] (Internal)
Low	31	SPI (Internal)

In addition to the above, the TX49/H3 core has a TX49/H3 core internal timer interrupt and two software interrupts, but these interrupts are directly reported to the TX49/H3 core independently of this Interrupt Controller. Please refer to the 64-bit TX System RISC TX49/H3 Core Architecture Manual for more information.

15.3.2 Interrupt request detection

In order to perform interrupt detection, each register of the Interrupt Controller is initialized, then the IDE bit of the Interrupt Detection Enable Register (IRDEN) is set to "1." All interrupts detected by the Interrupt Controller are masked when this bit is cleared.

It is possible to set each interrupt factor detection mode using Interrupt Detection Mode Register 0 (IRDM0) and Interrupt Detection Mode Register 1 (IRDM1). There are four detection modes: Low level, High level, falling edge, and rising edge.

The detected interrupt factors can be read out from the Interrupt Pending Register (IRPND).

15.3.3 Interrupt level assigning

Interrupt levels from 0 to 7 are assigned to each detected interrupt using the Interrupt Level Register (IRLVL0-7). Interrupt level 7 is the highest priority and interrupt level1 is the lowest priority. Level 0 interrupts will be masked. (Table 15.3.2).

The priorities set by these interrupt levels will be given higher priority than the priorities provided for each interrupt source indicated in Table 15.3.1.

Table 15.3.2 Interrupt Levels

Priority	Interrupt Level (IRLVLn.ILm)
High	111
	110
	101
	100
	011
	010
Low	001
Mask	000

15.3.4 Interrupt priority assigning

When multiple interrupt requests exist, the Interrupt Controller selects the interrupt with the highest priority according to the priority level and interrupt number. Interrupt factors with an interrupt level equal to or lower than the interrupt level specified by the Interrupt Mask Level Register (IRMSK) will be excluded (masked).

When the interrupt with the highest priority is selected, then the interrupt number of that interrupt is set in the interrupt factor field (CAUSE) of the Interrupt Current Status Register (IRCS), the interrupt level is set in the Interrupt Level field (LVL), and the Interrupt Flag bit (IF) is set.

Priorities are assigned as follows.

- When interrupt levels differ, the interrupt with the higher interrupt level has priority (Table 15.3.2)
- When multiple interrupts with the same interrupt level are simultaneously detected, the interrupt with the smaller interrupt number has priority (Table 15.3.1).

In the following cases, interrupts are reprioritized. If any new interrupt requests are generated before reprioritization, the highest-priority interrupt is accepted, changing the Interrupt Cause (CAUSE) and Interrupt Level (LVL) fields in the Interrupt Current Status (IRCS) register.

- When an interrupt request with a higher interrupt level than that of the currently selected interrupt is detected. If the interrupt levels are equal, the Interrupt Cause (CAUSE) field does not change, even if the interrupt number is smaller.
- When the interrupt level (IRLVLn.ILm) of the currently selected interrupt changes to a value smaller than the current setting.
- When the currently selected interrupt is cleared (refer to 15.3.6 Clearing interrupt requests).

Changing the Interrupt Mask Level (IRMSK.LML) does not cause the IRC to reprioritize interrupts. However, if the Interrupt Mask Level (IRMSK.LML) is set to a value equal to or greater than the current interrupt level (IRLVLn.ILm), the Interrupt Flag bit in the Interrupt Current Status register (IRCS.IF) is set to mask the interrupt.

15.3.5 Interrupt notification

When the interrupt with the highest priority is selected, then the interrupt factor is reported to the Interrupt Current Status Register (IRCS) and an interrupt is reported to the TX49/H3 core.

The TX49/H3 core distinguishes interrupt factors using the IP field (IP[7:2]) of the Cause Register. The interrupt notification from the Interrupt Controller is reflected in the IP[2] bit. The Interrupt Handler uses the IP[2] bit to judge whether or not there are interrupts from this Interrupt Controller and uses the Interrupt Current Status Register (IRCS) to determine the interrupt cause.

The Interrupt Factor field (IRCS.CAUSE) value is reflected in the remaining bits of the IP field. Since bit IP[7] is also being used for notification of TX49/H3 CPU core internal timer interrupts, the contents specified by IP[7] differ according to whether internal timer interrupts are set to valid (TINTDIS=0) or invalid (TINTDIS=1), as indicated Table 15.3.3.

TINTDIS is the value that is set from DATA[7] at the timing when the RESET* signal is deasserted. See the explanation “3.2 Boot Configuration” for more information.

Table 15.3.3 Interrupt Notification to IP[7:2] of the CP0 Cause Register

TINTDIS	IP[7]	IP[6:3]	IP[2]
0 (Internal Timer Interrupts: Valid)	Internal Timer Interrupt Notification	IRCS.CAUSE[3:0]	IRCS.IF
1 (Internal Timer Interrupts: Invalid)	IRCS.CAUSE[4:0]		IRCS.IF

15.3.6 Clearing interrupt requests

Interrupt requests are cleared according to the following process.

- When the detection mode is set to the High level or Low level:
  Operation is performed to deassert the request of a source that is asserting an interrupt request.

- When the detection mode is set to Rising edge or Falling edge
  Edge detection requests are cleared by first specifying the interrupt source of the interrupt request to be cleared in the Edge Detection Clear Source field (EDCS0 or EDCS1) of the Interrupt Edge Detection Clear Register (IREDC) then writing the resulting value when the corresponding Edge Detection Clear Enable bit (EDCE0 or EDCE1) is set to “1.”

15.3.7 Interrupt requests

It is possible to make interrupt requests to external devices and interrupt requests (IRC interrupts) to the TX49/H3 core by using a 16-bit interrupt request flag register. REQ[1]* signals are used as interrupt output signals. Consequently, external interrupt requests can only be used when in the PCI External Arbiter mode. Also, internal interrupt requests are assigned to interrupt number 13 of the Interrupt Controller (IRC).

The following six registers set the interrupts.

- Interrupt Request Flag Register (IRFLAG0, IRFLAG1)
- Interrupt Request Polarity Control Register (IRPOL)
- Interrupt Request Mask Register (IRMASKINT, IRMASKEXT)
- Interrupt Request Control Register (IRRCNT)

The following formulas derive the interrupt generation conditions:

Internal interrupt request =

(|((IRFLAG[15:0] ^ IRPOL[15:0]) & IRMASKINT[15:0]))^ IRRCNT.INTPOL

External interrupt request =

(|((IRFLAG[15:0] ^ IRPOL[15:0]) & IRMASKEXT[15:0]))^ IRRCNT.EXTPOL

In the above formulas, "^" indicates Exclusive OR operations and "|" indicates reduction operators that perform an OR operation on all bits.

Also, the External Interrupt OD Control bit (IRRCNT.OD) of the Interrupt Request Control Register can select whether the external interrupt supply signal is open drain output or totem pole output.

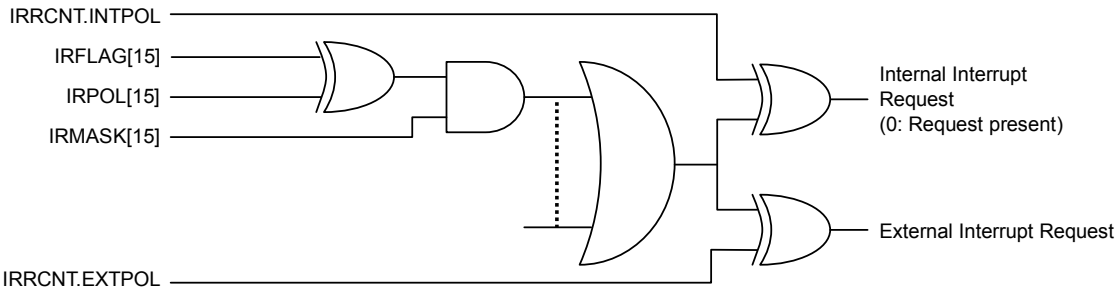


Figure 15.3.1 External Interrupt Request Logic

There are two flag registers: Flag Register 0 (IRFLAG0), and Flag Register 1 (IRFLAG1). These registers have two different Write methods. Accordingly, Writes to one register are reflects in the other.

Either “0” or “1” can be written to Flag Register 0

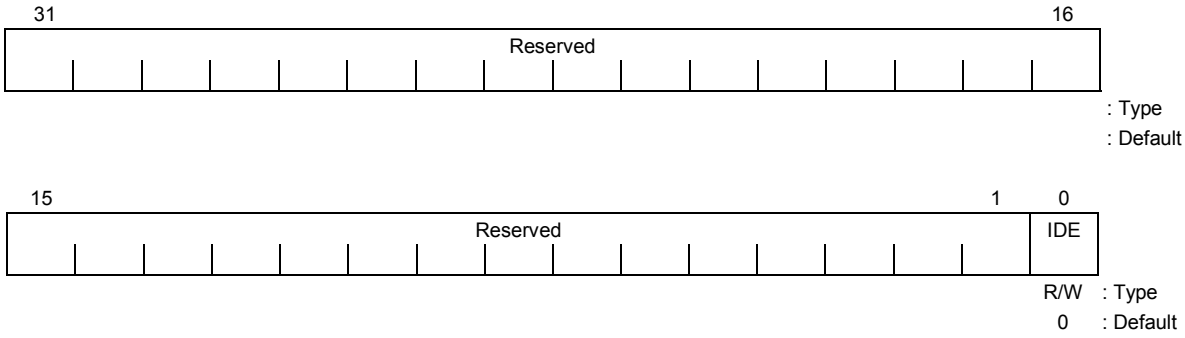
In the case of Flag Register 1 however, “1” can be written from the TX49/H3 core, but “0” cannot be written. On the other hand, bits that wrote “1” are cleared to “0” in the case of access from a device other than the TX49/H3 core (access from an external PCI device for example). The bit value at this time will not change even if “0” is written. This register sends interrupt notification from the TX49/H3 core to external devices. External devices can be used in applications that clear these interrupt notifications.

15.4 Registers

Table 15.4.1 Interrupt Control Registers

Address	Register	Register Name
0xF600	IRDEN	Interrupt Detection Enable Register
0xF604	IRDM0	Interrupt Detection Mode Register 0
0xF608	IRDM1	Interrupt Detection Mode Register 1
0xF610	IRLVL0	Interrupt Level Register 0
0xF614	IRLVL1	Interrupt Level Register 1
0xF618	IRLVL2	Interrupt Level Register 2
0xF61C	IRLVL3	Interrupt Level Register 3
0xF620	IRLVL4	Interrupt Level Register 4
0xF624	IRLVL5	Interrupt Level Register 5
0xF628	IRLVL6	Interrupt Level Register 6
0xF62C	IRLVL7	Interrupt Level Register 7
0xF640	IRMSK	Interrupt Mask Register
0xF660	IREDC	Interrupt Edge Detection Clear Register
0xF680	IRPND	Interrupt Pending Register
0xF6A0	IRCS	Interrupt Current Status Register
0xF510	IRFLAG0	Interrupt Request Flag Register 0
0xF514	IRFLAG1	Interrupt Request Flag Register 1
0xF518	IRPOL	Interrupt Request Polarity Control Register
0xF51C	IRRCNT	Interrupt Request Control Register
0xF520	IRMASKINT	Interrupt Request Internal Interrupt Mask Register
0xF524	IRMASKEXT	Interrupt Request External Interrupt Mask Register

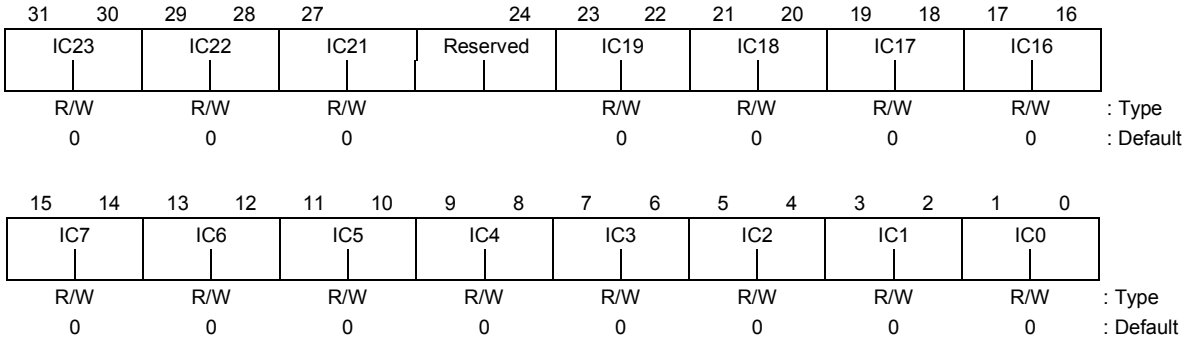
15.4.1 Interrupt Detection Enable Register (IRDEN) 0xF600



Bit(s)	Mnemonic	Field Name	Explanation	Read/Write
31:1	—	—	Reserved	—
0	IDE	Interrupt Control Enable	Interrupt Detection Enable (Default: 0) Enables interrupt detection. 0: Stop interrupt detection. 1: Start interrupt detection	R/W

Figure 15.4.1 Interrupt Detection Enable Register

15.4.2 Interrupt Detection Mode Register 0 (IRDM0) 0xF604


31 30 29 28 27 24 23 22 21 20 19 18 17 16
IC23 IC22 IC21 Reserved IC19 IC18 IC17 IC16
R/W R/W R/W R/W R/W R/W R/W : Type
0 0 0 0 0 0 0 : Default
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
IC7 IC6 IC5 IC4 IC3 IC2 IC1 IC0
R/W R/W R/W R/W R/W R/W R/W R/W : Type
0 0 0 0 0 0 0 0 : Default

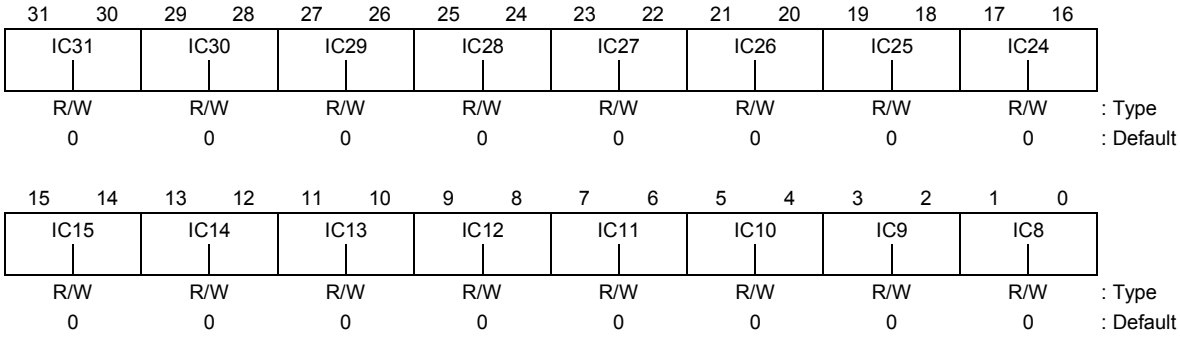
Bits	Mnemonic	Field Name	Explanation	Read/Write
31:30	IC23	Interrupt Source Control 23	Interrupt Source Control 23 (Default: 00) These bits specify the active state of PCIPMC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
29:28	IC22	Interrupt Source Control 22	Interrupt Source Control 22 (Default: 00) These bits specify the active state of PCIERR0 interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
27:26	IC21	Interrupt Source Control 21	Interrupt Source Control 21 (Default: 00) These bits specify the active state of NDFMC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
25:24	—	—	Reserved	—
23:22	IC19	Interrupt Source Control 19	Interrupt Source Control 19 (Default: 00) These bits specify the active state of TMR[2] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
21:20	IC18	Interrupt Source Control 18	Interrupt Source Control 18 (Default: 00) These bits specify the active state of TMR[1] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
19:18	IC17	Interrupt Source Control 17	Interrupt Source Control 17 (Default: 00) These bits specify the active state of TMR[0] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W

Figure 15.4.2 Interrupt Detection Mode Register 0 (1/2)

Bits	Mnemonic	Field Name	Explanation	Read/Write
17:16	IC16	Interrupt Source Control 16	Interrupt Source Control 16 (Default: 00) These bits specify the active state of PCIC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
15:14	IC7	Interrupt Source Control 7	Interrupt Source Control 7 (Default: 00) These bits specify the active state of external INT[5] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
13:12	IC6	Interrupt Source Control 6	Interrupt Source Control 6 (Default: 00) These bits specify the active state of external INT[4] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
11:10	IC5	Interrupt Source Control 5	Interrupt Source Control 5 (Default: 00) These bits specify the active state of external INT[3] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
9:8	IC4	Interrupt Source Control 4	Interrupt Source Control 4 (Default: 00) These bits specify the active state of external INT[2] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
7:6	IC3	Interrupt Source Control 3	Interrupt Source Control 3 (Default: 00) These bits specify the active state of external INT[1] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
5:4	IC2	Interrupt Source Control 2	Interrupt Source Control 2 (Default: 00) These bits specify the active state of external INT[0] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
3:2	IC1	Interrupt Source Control 1	Interrupt Source Control 1 (Default: 00) These bits specify the active state of TX49 Write Timeout Error interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
1:0	IC0	Interrupt Source Control 0	Interrupt Source Control 0 (Default: 00) These bits specify the active state of ECC Error interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W

Figure 15.4.2 Interrupt Detection Mode Register 0 (2/2)

15.4.3 Interrupt Detection Mode Register 1 (IRDM1) 0xF608



Bit	Mnemonic	Field Name	Explanation	Read/Write
31:30	IC31	Interrupt Source Control 31	Interrupt Source Control 31 (Default: 00, R/W) These bits specify the active state of SPI interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
29:28	IC30	Interrupt Source Control 30	Interrupt Source Control 30 (Default: 00, R/W) These bits specify the active state of DMA1[3] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
27:26	IC29	Interrupt Source Control 29	Interrupt Source Control 29 (Default: 00, R/W) These bits specify the active state of DMA1[2] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
25:24	IC28	Interrupt Source Control 28	Interrupt Source Control 28 (Default: 00, R/W) These bits specify the active state of DMA1[1] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
23:22	IC27	Interrupt Source Control 27	Interrupt Source Control 27 (Default: 00, R/W) These bits specify the active state of DMA1[0] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
21:20	IC26	Interrupt Source Control 26	Interrupt Source Control 26 (Default: 00, R/W) These bits specify the active state of PCIC1 interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W

Figure 15.4.3 Interrupt Detection Mode Register 1 (1/3)

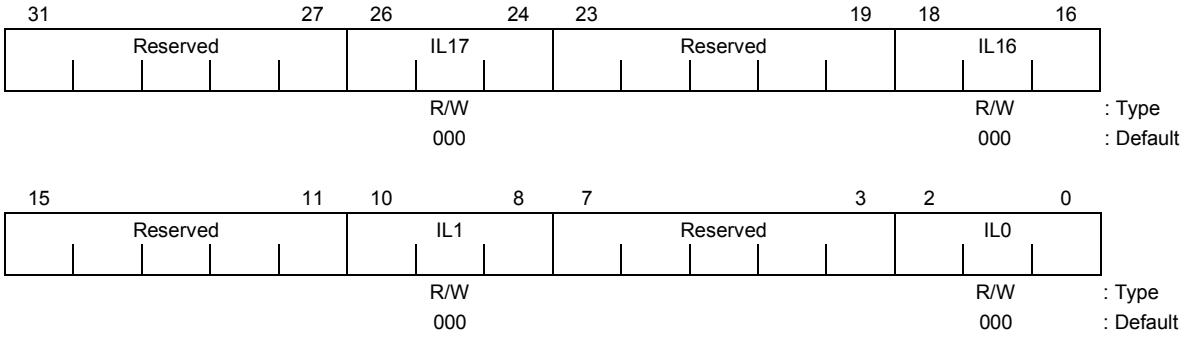
Bit	Mnemonic	Field Name	Explanation	Read/Write
19:18	IC25	Interrupt Source Control 25	Interrupt Source Control 25 (Default: 00, R/W) These bits specify the active state of ACLCPME interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
17:16	IC24	Interrupt Source Control 24	Interrupt Source Control 24 (Default: 00, R/W) These bits specify the active state of ACLC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
15:14	IC15	Interrupt Source Control 15	Interrupt Source Control 15 (Default: 00) These bits specify the active state of PDMAC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
13:12	IC14	Interrupt Source Control 14	Interrupt Source Control 14 (Default: 00) These bits specify the active state of IRC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
11:10	IC13	Interrupt Source Control 13	Interrupt Source Control 13 (Default: 00) These bits specify the active state of DMA0 [3] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
9:8	IC12	Interrupt Source Control 12	Interrupt Source Control 12 (Default: 00) These bits specify the active state of DMA0 [2] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
7:6	IC11	Interrupt Source Control 11	Interrupt Source Control 11 (Default: 00) These bits specify the active state of DMA0 [1] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
5:4	IC10	Interrupt Source Control 10	Interrupt Source Control 10 (Default: 00) These bits specify the active state of DMA0 [0] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W

Figure 15.4.3 Interrupt Detection Mode Register 1 (2/3)

Bit	Mnemonic	Field Name	Explanation	Read/Write
3:2	IC9	Interrupt Source Control 9	Interrupt Source Control 9 (Default: 00) These bits specify the active state of SIO[1] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W
1:0	IC8	Interrupt Source Control 8	Interrupt Source Control 8 (Default: 00) These bits specify the active state of SIO[0] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable	R/W

Figure 15.4.3 Interrupt Detection Mode Register 1 (3/3)

15.4.4 Interrupt Level Register 0 (IRLVL0) 0xF610



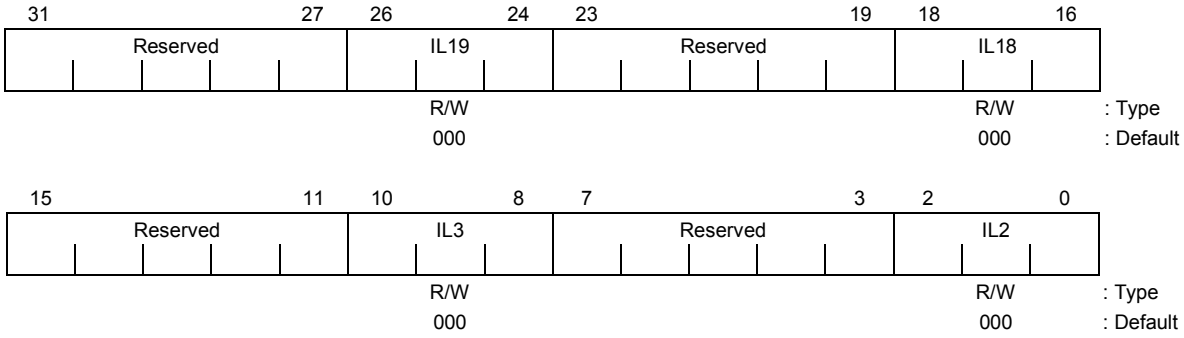
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL17	Interrupt Level 17	Interrupt Level of INT [17] (Default: 000) These bits specify the interrupt level of [TMR [0] 000: Interrupt Level 0 (Interrupt disable) 001:Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL16	Interrupt Level 16	Interrupt Level of INT [16] (Default: 000) These bits specify the interrupt level of PCIC0 interrupts. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
15:11	—	—	Reserved	—
10:8	IL1	Interrupt Level 1	Interrupt Level of INT [1] (Default: 000) These bits specify the interrupt level for TX49 Write Timeout Error interrupts. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010:Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—

Figure 15.4.4 Interrupt Level Register 0 (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
2:0	IL0	Interrupt Level 0	Interrupt Level of INT [0] (Default: 000) These bits specify the interrupt level of ECC Error interrupts. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.5 Interrupt Level Register 0 (2/2)

15.4.5 Interrupt Level Register (IRLVL1) 0xF614



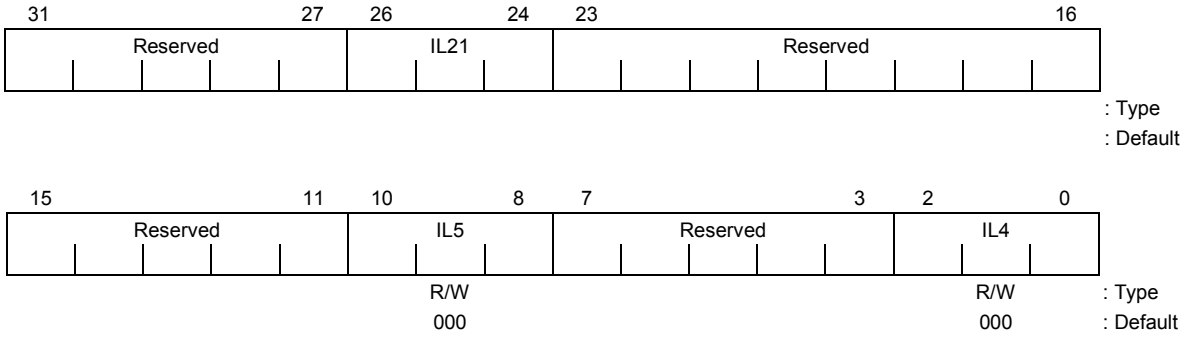
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL19	Interrupt Level 19	Interrupt Level of INT [19] (Default: 000) These bits specify the interrupt level of TMR [2]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL18	Interrupt Level 18	Interrupt Level of INT [18] (Default: 000) These bits specify the interrupt level of TMR[1]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
15:11	—	—	Reserved	—
10:8	IL3	Interrupt Level 3	Interrupt Level of INT [3] (Default: 000) These bits specify the interrupt level of external INT[1]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—

Figure 15.4.5 Interrupt Level Register 1 (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
2:0	IL2	Interrupt Level 2	Interrupt Level of INT [2] (Default: 000) These bits specify the interrupt level of external INT[0]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.5 Interrupt Level Register 1 (2/2)

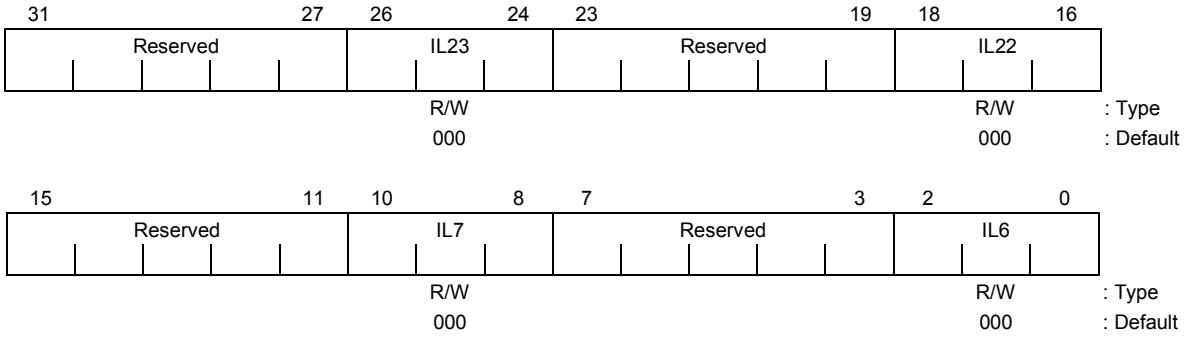
15.4.6 Interrupt Level Register 2 (IRLVL2) 0xF618



Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL21	Interrupt Level 21	Interrupt Level of INT [21] (Default: 000) These bits specify the interrupt level of NDFMC. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:11	—	—	Reserved	—
10:8	IL5	Interrupt Level 5	Interrupt Level of INT [5] (Default: 000) These bits specify the interrupt level of external INT[3] / ETHERC1. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—
2:0	IL4	Interrupt Level 4	Interrupt Level of INT [4] (Default: 000) These bits specify the interrupt level of external INT[2]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.6 Interrupt Level Register 2

15.4.7 Interrupt Level Register 3 (IRLVL3) 0xF61C



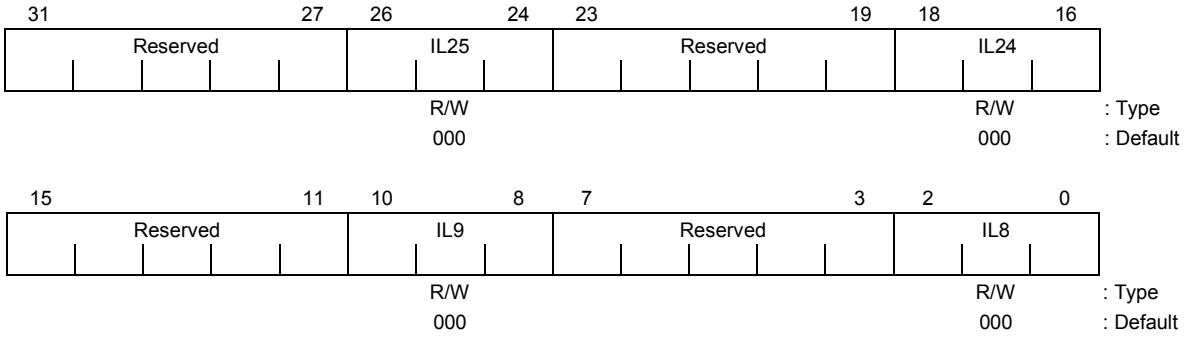
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL23	Interrupt Level 23	Interrupt Level of INT [23] (Default: 000) These bits specify the interrupt level of PCIPME0 interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL22	Interrupt Level 22	Interrupt Level of INT [22] (Default: 000) These bits specify the interrupt level of PCIERR0 interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
15:11	—	—	Reserved	—
10:8	IL7	Interrupt level 7	Interrupt Level of INT [7] (Default: 000) These bits specify the interrupt level of external INT[5]. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—

Figure 15.4.7 Interrupt Level Register 3 (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
2:0	IL6	Interrupt level 6	Interrupt Level of INT [6] (Default: 000) These bits specify the interrupt level of external INT[4] / ETHERC0. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.7 Interrupt Level Register 3 (2/2)

15.4.8 Interrupt Level Register 4 (IRLVL4) 0xF620



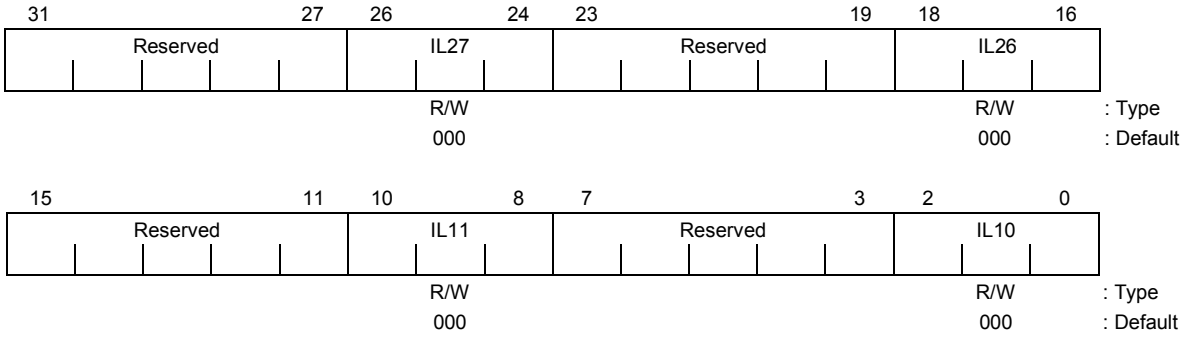
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL25	Interrupt level 25	Interrupt Level of INT [25] (Default: 000, R/W) These bits specify the interrupt level of ACLCPME interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL24	Interrupt level 24	Interrupt Level of INT [24] (Default: 000, R/W) These bits specify the interrupt level of ACLC interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
15:11	—	—	Reserved	—
10:8	IL9	Interrupt level 9	Interrupt Level of INT [9] (Default: 000) These bits specify the interrupt level of SIO [1] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—

Figure 15.4.8 Interrupt Level Register 4 (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
2:0	IL8	Interrupt level 8	Interrupt Level of INT [8] (Default: 000) These bits specify the interrupt level of SIO [0] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.8 Interrupt Level Register 4 (2/2)

15.4.9 Interrupt Level Register 5 (IRLVL5) 0xF624



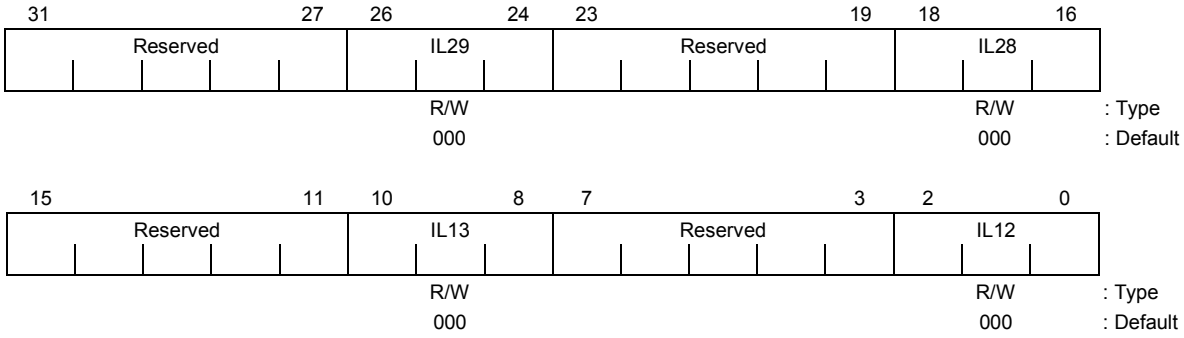
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL27	Interrupt level 27	Interrupt Level of INT [27] (Default: 000) These bits specify the interrupt level of DMA1 [0] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL26	Interrupt level 26	Interrupt Level of INT [26] (Default: 000) These bits specify the interrupt level of PCIC1 interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
15:11	—	—	Reserved	—
10:8	IL11	Interrupt level 11	Interrupt Level of INT [11] (Default: 000) These bits specify the interrupt level of DMA0 [1] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—

Figure 15.4.9 Interrupt Level Register 5 (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
2:0	IL10	Interrupt Level 10	Interrupt Level of INT [10] (Default: 000) These bits specify the interrupt level of DMA0 [0] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.9 Interrupt Level Register 5 (2/2)

15.4.10 Interrupt Level Register 6 (IRLVL6) 0xF628



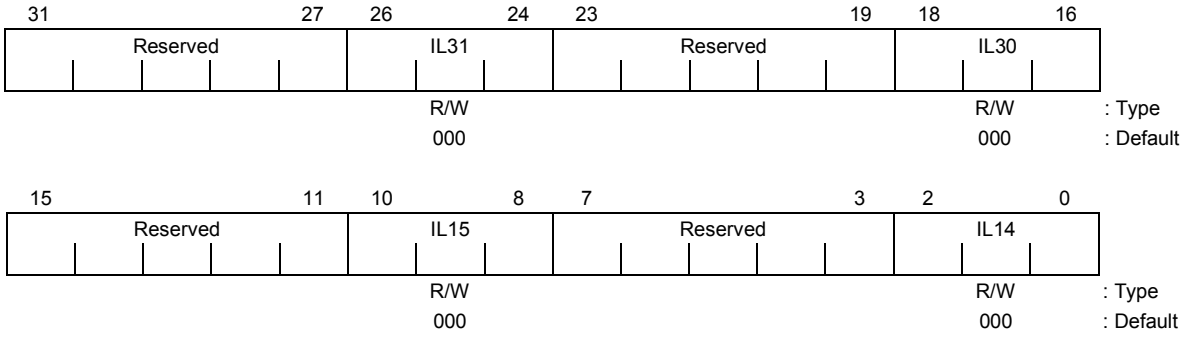
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL29	Interrupt level 29	Interrupt Level of INT [29] (Default: 000) These bits specify the interrupt level of DMA1 [2] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL28	Interrupt level 28	Interrupt Level of INT [28] (Default: 000) These bits specify the interrupt level of DMA1 [1] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
15:11	—	—	Reserved	—
10:8	IL13	Interrupt level 13	Interrupt Level of INT [13] (Default: 000) These bits specify the interrupt level of DMA0 [3] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—

Figure 15.4.10 Interrupt Level Register 6 (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
2:0	IL12	Interrupt level 12	Interrupt Level of INT [12] (Default: 000) These bits specify the interrupt level of DMA0 [2] interrupts. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.10 Interrupt Level Register 6 (2/2)

15.4.11 Interrupt Level Register 7 (IRLVL7) 0xF62C



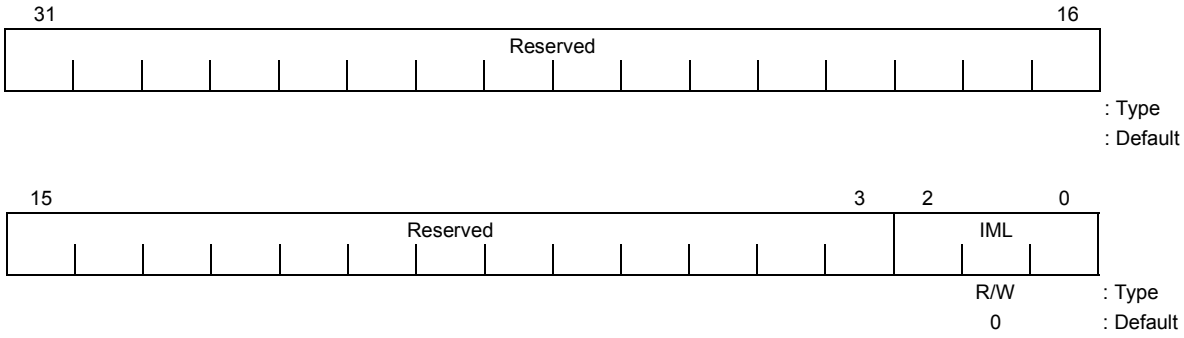
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL31	Interrupt level 31	Interrupt Level of INT [31] (Default: 000) These bits specify the interrupt level of SPI interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL31	Interrupt level 30	Interrupt Level of INT [30] (Default: 000) These bits specify the interrupt level of DMA1[3] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
15:11	—	—	Reserved	—
10:8	IL15	Interrupt level 15	Interrupt Level of INT [15] (Default: 000) These bits specify the interrupt level of PDMAC interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
7:3	—	—	Reserved	—

Figure 15.4.11 Interrupt Level Register 7 (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
2:0	IL14	Interrupt Level 14	Interrupt Level of INT [14] (Default: 000) These bits specify the interrupt level of IRC interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W

Figure 15.4.11 Interrupt Level Register 7 (2/2)

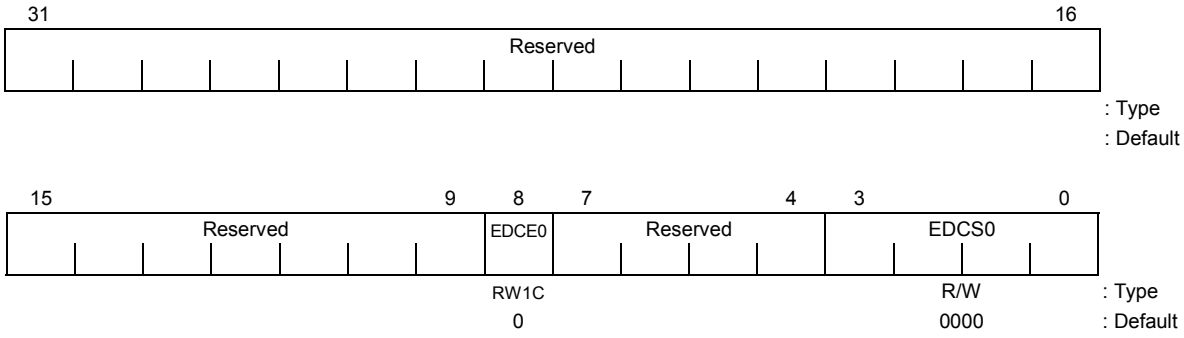
15.4.12 Interrupt Mask Level Register (IRMSK) 0xF640



Bit	Mnemonic	Field Name	Explanation	Read/Write
31:3	—	—	Reserved	—
2:0	IML	Interrupt Mask Level	Interrupt Mask Level (Default: 000) These bits specify the interrupt mask level. Masks interrupts with a mask level equal to or lower than the set mask level. 000: Interrupt mask level 0 (No interrupts masked) 001: Interrupt mask level 1 (Levels 2-7 enabled) 010: Interrupt mask level 2 (Levels 3-7 enabled) 011: Interrupt mask level 3 (Levels 4-7 enabled) 100: Interrupt mask level 4 (Levels 5-7 enabled) 101: Interrupt mask level 5 (Levels 6-7 enabled) 110: Interrupt mask level 6 (Level 7 enabled) 111: Interrupt mask level 7 (Interrupts disabled)	R/W

Figure 15.4.12 Interrupt Mask Register

15.4.13 Interrupt Edge Detection Clear Register (IREDC) 0xF660



Bit	Mnemonic	Field Name	Explanation	Read/Write
31:9	—	—	Reserved	—
8	EDCE0	Edge Detection Clear Enable 0	Edge Detection Clear Enable 0 (Default: 0) Clears edge detection of interrupts specified by the EDCS0 field. 0: Does not clear. 1: Clears. Value always becomes “0” when this bit is read.	R/W1C
7:4	—	—	Reserved	—
3:0	EDCS0	Edge Detection Clear Source 0	Edge Detection Clear Source 0 (Default: 0x0) These bits specify the interrupt source to be cleared. 1111: Reserved 1110: Reserved 1101: Reserved 1100: Reserved 1011: Reserved 1010: Reserved 1001: Reserved 1000: Reserved 0111: External INT [5] interrupt 0110: External INT [4] interrupt 0101: External INT [3] interrupt 0100: External INT [2] interrupt 0011: External INT [1] interrupt 0010: External INT [0] interrupt 0001: Reserved 0000: Reserved	R/W

Figure 15.4.13 Interrupt Status Control Register

15.4.14 Interrupt Pending Register (IRPND) 0xF680

Indicates the status of each interrupt request regardless of the IRLVL 7-0 and IRMSK value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	Reserved	IS19	IS18	IS17	IS16	
R	R	R	R	R	R	R	R	R	R	R		R	R	R	R	: Type
0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	: Default

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	: Type
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	: Default

Bit	Mnemonic	Field Name	Explanation	Read/Write
31	IS31	Interrupt Status 31	IRINTREQ [31] Status (Default: 0, R) This bit indicates the SPI interrupt status. 1: Interrupt requests 0: No interrupt requests	R
30	IS30	Interrupt Status 30	IRINTREQ [30] Status (Default: 0, R) This bit indicates the DMA1[3] interrupt status. 1: Interrupt requests 0: No interrupt requests	R
29	IS29	Interrupt Status 29	IRINTREQ [29] Status (Default: 0, R) This bit indicates the DAM1[2] interrupt status. 1: Interrupt requests 0: No interrupt requests	R
28	IS28	Interrupt Status 28	IRINTREQ [28] Status (Default: 0, R) This bit indicates the DMA1[1] interrupt status. 1: Interrupt requests 0: No interrupt requests	R
27	IS27	Interrupt Status 27	IRINTREQ [27] Status (Default: 0, R) This bit indicates the DMA1[0] interrupt status. 1: Interrupt requests 0: No interrupt requests	R
26	IS26	Interrupt Status 26	IRINTREQ [26] Status (Default: 0, R) This bit indicates the PCIC1 interrupt status. 1: Interrupt requests 0: No interrupt requests	R
25	IS25	Interrupt Status 25	IRINTREQ [25] Status (Default: 0, R) This bit indicates the ACLCPME interrupt status. 1: Interrupt requests 0: No interrupt requests	R
24	IS24	Interrupt Status 24	IRINTREQ [24] Status (Default: 0, R) This bit indicates the ACLC interrupt status. 1: Interrupt requests 0: No interrupt requests	R
23	IS23	Interrupt Status 23	IRINTREQ [23] status This bit indicates the PCIPMC interrupt status 1: Interrupt requests 0: No interrupt requests	R

Figure 15.4.14 Interrupt Source Status Register (1/3)

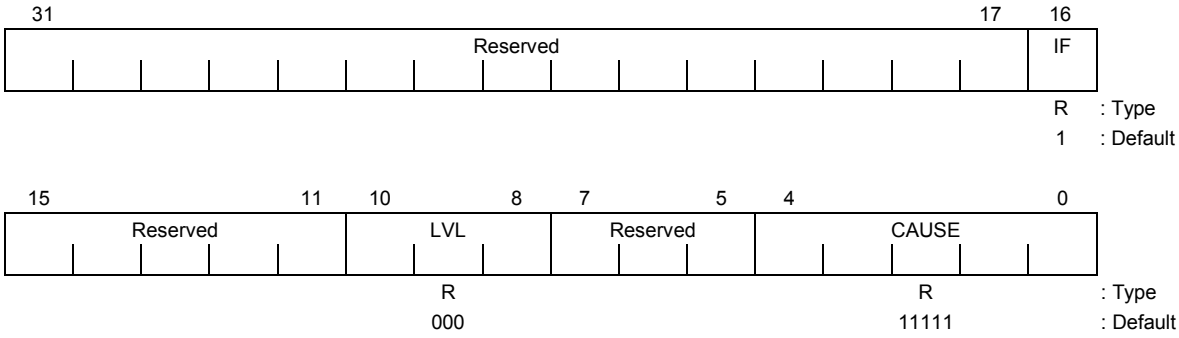
Bit	Mnemonic	Field Name	Explanation	Read/Write
22	IS22	Interrupt Status 22	IRINTREQ [22] status This bit indicates the PCIERR error status. 1: Interrupt requests 0: No interrupt requests	R
21	IS21	Interrupt Status 21	IRINTREQ [21] status This bit indicates the NDFMC interrupt status. 1: Interrupt requests 0: No interrupt requests	R
20	—	—	Reserved	—
19	IS19	Interrupt Status 19	IRINTREQ [19] status This bit indicates the TMR [2] interrupt status. 1: Interrupt requests 0: No interrupt requests	R
18	IS18	Interrupt Status 18	IRINTREQ [18] status This bit indicates the TMR [1] interrupt status. 1: Interrupt requests 0: No interrupt requests	R
17	IS17	Interrupt Status 17	IRINTREQ [17] status This bit indicates the TMR[0] interrupt status 1: Interrupt requests 0: No interrupt requests	R
16	IS16	Interrupt Status 16	IRINTREQ [16] status This bit indicates the PCIC interrupt status 1: Interrupt requests 0: No interrupt requests	R
15	IS15	Interrupt Status 15	IRINTREQ [15] status This bit indicates the PDMAC interrupt status. 1: Interrupt requests 0: No interrupt requests	R
14	IS14	Interrupt Status 14	IRINTREQ [14] status This bit indicates the IRC interrupt status 1: Interrupt requests 0: No interrupt requests	R
13	IS13	Interrupt Status 13	IRINTREQ [13] status This bit indicates the DMA [3] interrupt status 1: Interrupt requests 0: No interrupt requests	R
12	IS12	Interrupt Status 12	IRINTREQ [12] status This bit indicates the status of DMA [2] interrupts. 1: Interrupt requests 0: No interrupt requests	R
11	IS11	Interrupt Status 11	IRINTREQ [11] status This bit indicates the status of DMA [1] interrupts. 1: Interrupt requests 0: No interrupts requests	R
10	IS10	Interrupt Status 10	IRINTREQ [10] status This bit indicates the status of DMA [0] interrupts. 1: Interrupt requests 0: No interrupt requests	R

Figure 15.4.14 Interrupt Source Status Register (2/3)

Bit	Mnemonic	Field Name	Explanation	Read/Write
9	IS9	Interrupt Status 9	IRINTREQ [9] status This bit indicates the status of SIO [1] interrupts. 1: Interrupt requests 0: No interrupt requests	R
8	IS8	Interrupt Status 8	IRINTREQ [8] status This bit indicates the status of SIO [0] interrupts. 1: Interrupt requests 0: No interrupt requests	R
7	IS7	Interrupt Status 7	IRINTREQ [7] status This bit indicates the status of external INT [5] interrupts. 1: Interrupt requests 0: No interrupt requests	R
6	IS6	Interrupt Status 6	IRINTREQ [6] status This bit indicates the status of external INT [4] interrupts. 1: Interrupt requests 0: No interrupt requests	R
5	IS5	Interrupt Status 5	IRINTREQ [5] status This bit indicates the status of external INT [3] interrupts. 1: Interrupt requests 0: No interrupt requests	R
4	IS4	Interrupt Status 4	IRINTREQ [4] status This bit indicates the status of external INT [2] interrupts. 1: Interrupt requests 0: No interrupt requests	R
3	IS3	Interrupt Status 3	IRINTREQ [3] status This bit indicates the status of external INT [1] interrupts. 1: Interrupt requests 0: No interrupt requests	R
2	IS2	Interrupt Status 2	IRINTREQ [2] status This bit indicates the status of external INT [0] interrupts. 1: Interrupt requests 0: No interrupt requests	R
1	IS1	Interrupt Status 1	IRINTREQ [1] status This bit indicates the status of TX49 Write Timeout Error interrupts. 1: Interrupt requests 0: No interrupt requests	R
0	IS0	Interrupt Status 0	IRINTREQ [0] status This bit indicates the status of ECC Error interrupts. 1: Interrupt requests 0: No interrupt requests	R

Figure 15.4.14 Interrupt Source Status Register (3/3)

15.4.15 Interrupt Current Status Register (IRCS) 0xF6A0



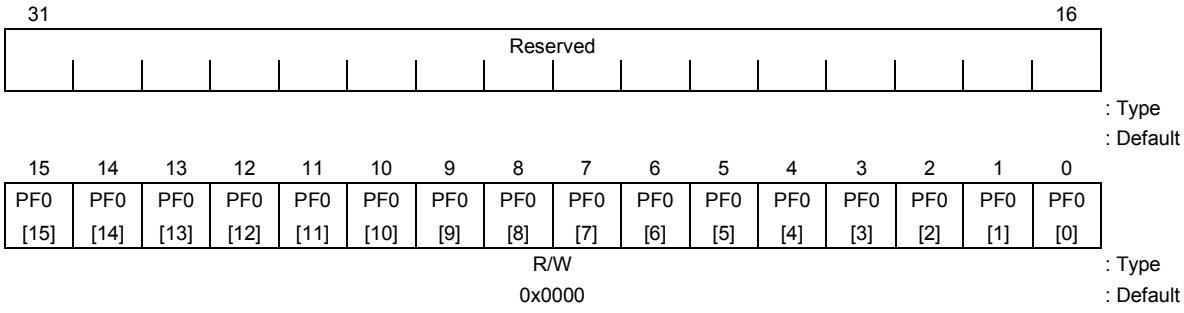
Bit	Mnemonic	Field Name	Explanation	Read/Write
31:17	—	—	Reserved	—
16	IF	Interrupt Flag	Interrupt Flag (Default: 1) This bit indicates the interrupt generation status. 0: Interrupt requests have been generated. 1: Interrupt requests have not been generated	R
15:11	—	—	Reserved	—
10:8	LVL	Interrupt Level	Interrupt Level (Default: 000) These bits specify the level of the interrupt request that was reported to the TX49/H3 core. This field becomes undefined if no interrupt request is pending (i.e., the IF bit is set). 000: Interrupt level 0 001: Interrupt level 1 : : 111: Interrupt level 7	R
7:5	—	—	Reserved	—

Figure 15.4.15 Interrupt Current Status Register (1/2)

Bit	Mnemonic	Field Name	Explanation	Read/Write
4:0	CAUSE	Interrupt Cause	Interrupt Cause (Default: 0x1F) These bits specify the interrupt cause that was reported to the TX49/H3 core. This field becomes undefined if no interrupt request is pending (i.e., the IF bit is set). 00000: ECC Error 00001: TX49 Write Timeout Error 00010: External INT [0] interrupt 00011: External INT [1] interrupt 00100: External INT [2] interrupt 00101: External INT [3] interrupt 00110: External INT [4] interrupt 00111: External INT [5] interrupt 01000: SIO [0] interrupt 01001: SIO [1] interrupt 01010: DMA0 [0] interrupt 01011: DMA0 [1] interrupt 01100: DMA0 [2] interrupt 01101: DMA0 [3] interrupt 01110: IRC interrupt 01111: PDMAC0 interrupt 10000: PCIC0 interrupt 10001: TMR [0] interrupt 10010: TMR [1] interrupt 10011: TMR [2] interrupt 10100: (Reserved) 10101: NDFMC interrupt 10110: PCIERR interrupt 10111: PCIPME interrupt 11000: ACLC interrupt 11001: ACLCPME interrupt 11010: PCIC1 interrupt 11011: DMA1[0] interrupt 11100: DMA1[1] interrupt 11101: DMA1[2] interrupt 11110: DMA1[3] interrupt 11111: SPI interrupt	R

Figure 15.4.15 Interrupt Current Status Register (2/2)

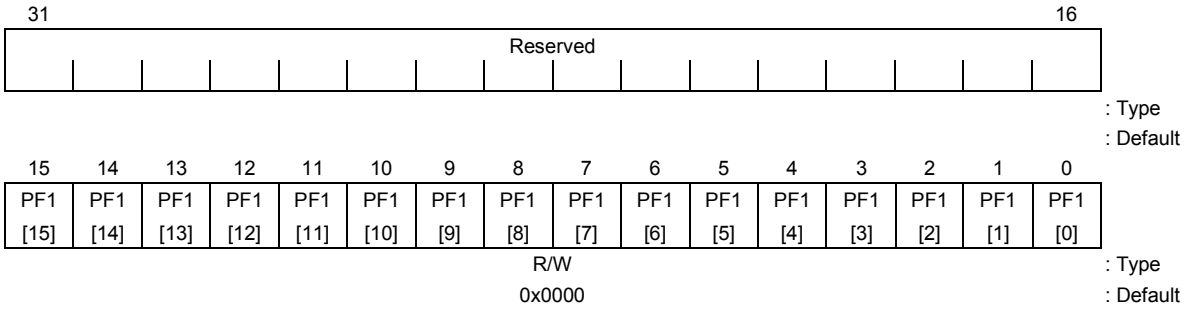
15.4.16 Interrupt Request Flag Register 0 (IRFLAG0) 0xF510



Bit	Mnemonic	Field Name	Explanation	Read/Write
31:16	—	—	Reserved	—
15:0	PF0 [15:0]	Flag 0	Interrupt Request Flag 0 [15:0] (Default: 0x0000) Changes made to this register are reflected in Flag Register 1 also since they are the same registers. The bits in this field accept writes of both 1s and 0s.	R/W

Figure 15.4.16 Interrupt Request Flag Register 0

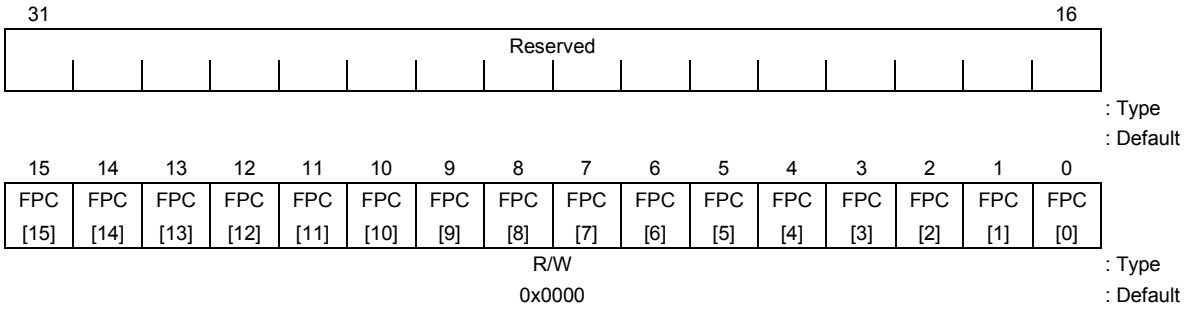
15.4.17 Interrupt Request Flag Register 1 (IRFLAG1) 0xF514



Bit	Mnemonic	Field Name	Explanation	Read/Write
31:16	—	—	Reserved	—
15:0	PF1 [15:0]	Flag 1	Interrupt Request Flag 1 [15:0] (Default: 0x0000) Changes made to this register are reflected in Flag Register 0 also since they are the same registers. Writes to Flag Register 1 operate as follows: Write Write from the TX49/H3 core 1: Set the flag bit 0: No change Write from other devices (DMAC, PCIC) 1: Clear the flag bit 0: No change Read: Read the flag bit	R/W

Figure 15.4.17 Interrupt Request Flag Register 1

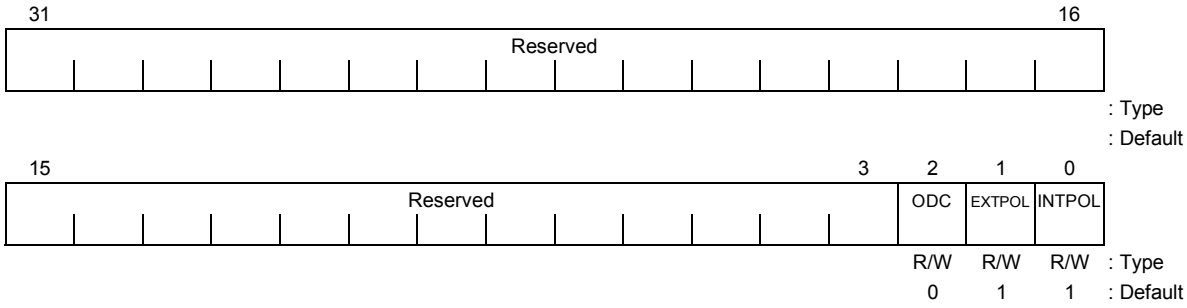
15.4.18 Interrupt Request Polarity Control Register (IRPOL) 0xF518


31
16
Reserved
: Type
: Default
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
FPC [15] FPC [14] FPC [13] FPC [12] FPC [11] FPC [10] FPC [9] FPC [8] FPC [7] FPC [6] FPC [5] FPC [4] FPC [3] FPC [2] FPC [1] FPC [0]
R/W
: Type
0x0000
: Default

Bit	Mnemonic	Field Name	Explanation	Read/Write
31:16	—	—	Reserved	—
15:0	FPC [15:0]	Flag Polarity Control	Flag Polarity Control [15:0] (Default: 0x0000) These bits specify the polarity of the flag bit that generated the interrupt. An interrupt request is generated when the XOR of the FPC bit and the flag bit is "1." Flag bit (PF) / FPC bit / Interrupt request 0 / 0 / No 0 / 1 / Yes 1 / 0 / Yes 1 / 1 / No	R/W

Figure 15.4.18 Interrupt Requests Polarity Control Register

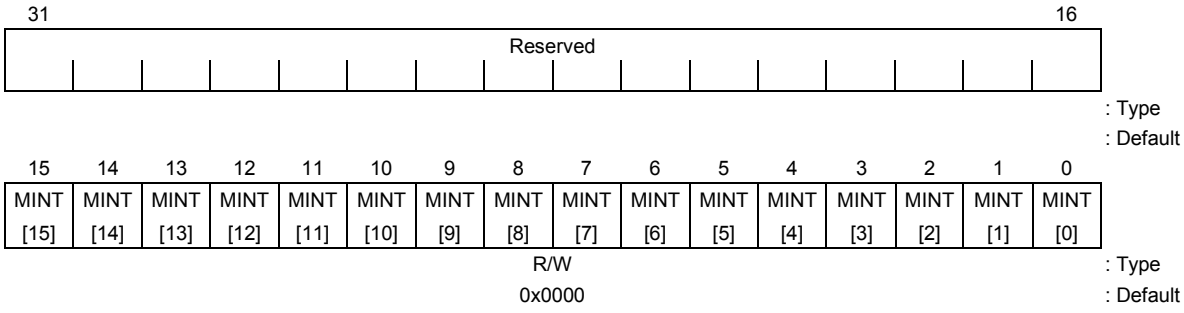
15.4.19 Interrupt Request Control Register (IRRCNT) 0xF51C



Bit	Mnemonic	Field Name	Explanation	Read/Write
31:3	—	—	Reserved	—
2	ODC	External Interrupt OD Control	External Interrupt Open Drain Control (Default: 0) This bit specifies whether to make the external interrupt signal (IRC[2]*) an open drain pin or not. 0: Open drain (reset) 1: Totem pole	R/W
1	EXTPOL	External Interrupt Request Polarity Control	External Interrupt Polarity Control (Default: 1) This bit specifies the polarity of external interrupt requests. 0: Do not reverse polarity of interrupt requests. 1: Reverse polarity of interrupt requests	R/W
0	INTPOL	Internal Interrupt Request Polarity Control	Internal Interrupt Polarity Control (Default: 1) This bit specifies the polarity of internal interrupt requests. 0: Do not reverse polarity of interrupt requests. 1: Reverse polarity of interrupt requests	R/W

Figure 15.4.19 Interrupt Request Control Register

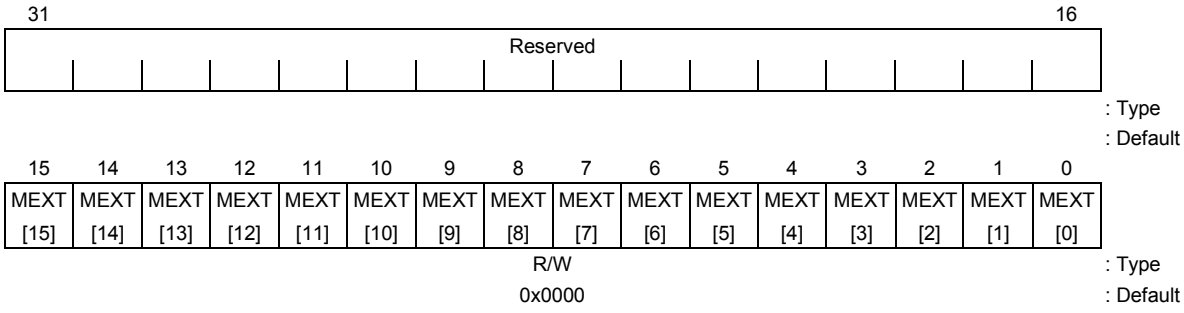
15.4.20 Interrupt Request Internal Interrupt Mask Register (IRMASKINT) 0xF520


31
16
Reserved
: Type
: Default
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MINT [15] MINT [14] MINT [13] MINT [12] MINT [11] MINT [10] MINT [9] MINT [8] MINT [7] MINT [6] MINT [5] MINT [4] MINT [3] MINT [2] MINT [1] MINT [0]
R/W
: Type
0x0000
: Default

Bit	Mnemonic	Field Name	Explanation	Read/Write
31:16	—	—	Reserved	—
15:0	MINT [15:0]	Internal Request Mask	Internal Interrupt Mask (Default: 0x0000) These bits specify whether to use the corresponding flag bit as an internal interrupt cause. Interrupt causes are masked when this bit is “0.” 0: Mask (Reset) 1: Do not mask	R/W

Figure 15.4.20 Interrupt Request Internal Interrupt Mask Register

15.4.21 Interrupt Request External Interrupt Mask Register (IRMASKEXT) 0xF524


31
16
Reserved
: Type
: Default
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MEXT [15] MEXT [14] MEXT [13] MEXT [12] MEXT [11] MEXT [10] MEXT [9] MEXT [8] MEXT [7] MEXT [6] MEXT [5] MEXT [4] MEXT [3] MEXT [2] MEXT [1] MEXT [0]
R/W
: Type
0x0000
: Default

Bit	Mnemonic	Field Name	Explanation	Read/Write
31:16	—	—	Reserved	—
15:0	MEXT [15:0]	External Request Mask	External Interrupt Mask (Default: 0x0000) These bits specify whether to use the corresponding flag bit as an external interrupt cause. Interrupt causes are masked when this bit is “0.” 0: Mask (reset) 1: Do not mask	R/W

Figure 15.4.21 Interrupt Request External Interrupt Mask Register

16. Ethernet Controller

16.1 Features

Mounted in the TX4938 is a 2-channel 10/100 Mbps Ethernet Controller. This Ethernet Controller has circuits equivalent to the Toshiba TC35815CF (but partially modified) built in.

16.2 Block diagram

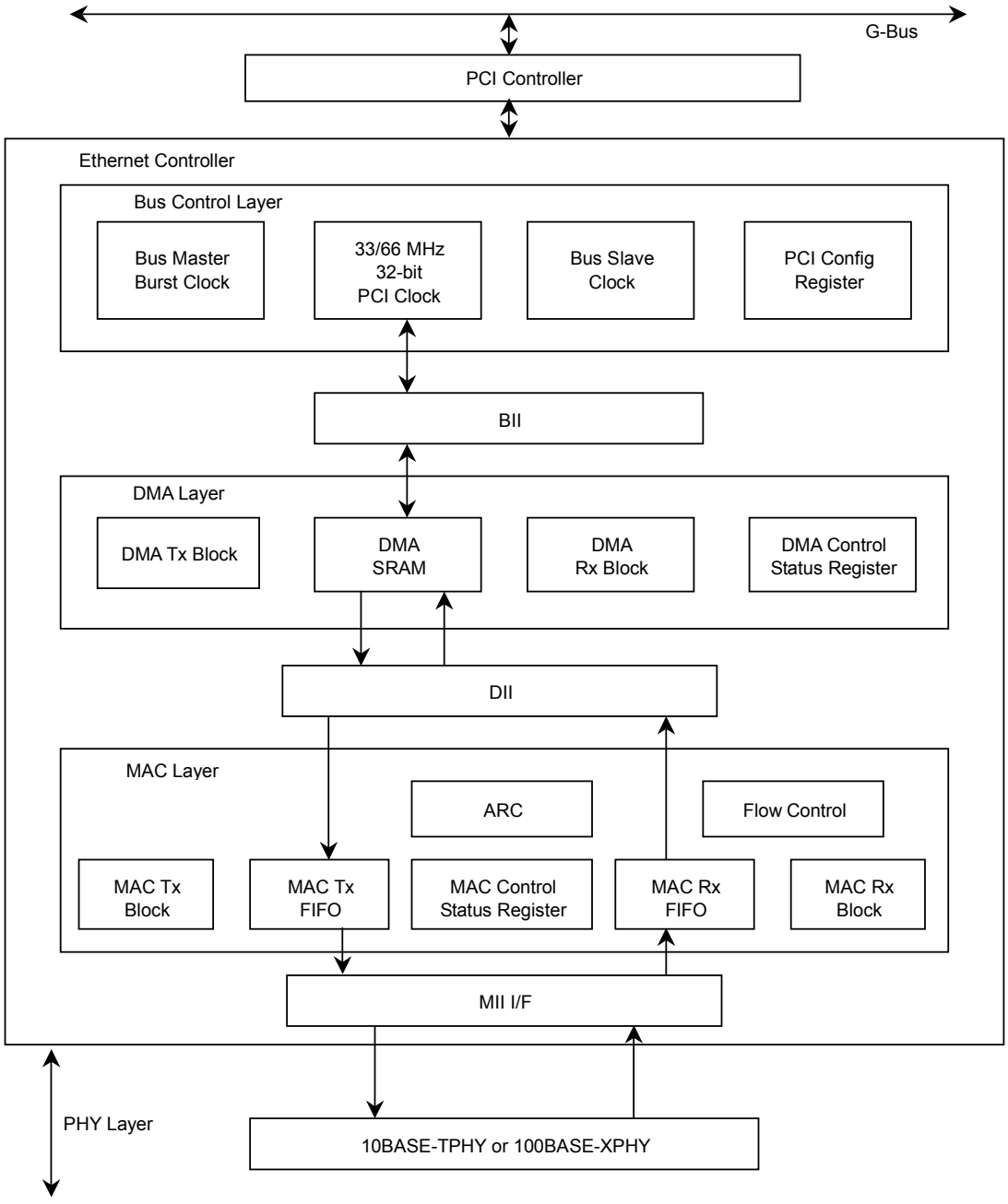


Figure 16.2.1 Ethernet Controller Block Diagram

16.3 Detailed explanation

16.3.1 Accessing the Ethernet Controller

The Ethernet Controller is connected to the chip-internal PCI Bus. The chip-internal PCI Bus is connected to the G-Bus via the PCI Controller (PCIC1). Therefore, you access the Ethernet Controller via PCI Controller 1.

PCI Controller 1 is a circuit that is equivalent to the PCI Controller (PCIC0) in Chapter 10. The Ethernet Controller only supports the PCI Controller 1 functions that it requires to operate. Following are the differences with PCIC0.

- Host mode only. (Does not support the Satellite mode.)
- Does not support EEPROM for Config data storage
- Does not support an external PCI arbiter
- Does not support type 1 configuration
- Does not support the Interrupt Acknowledge command
- Does not support the Special Cycle command

16.3.1.1 PCI Controller 1 control registers

Table 16.3.1 lists the PCI Controller 1 registers. For the details of each register or the operation of the PCI Controller, see Chapter 10.

Table 16.3.1 PCI Controller Control Registers (1/2)

Reference	Address	Size	Mnemonic	Register Name
10.4.1	0x7000	32	PCIID	ID Register (Device ID, Vendor ID)
10.4.2	0x7004	32	PCISTATUS	PCI Status, Command Register
10.4.3	0x7008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
10.4.4	0x700C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
10.4.5	0x7010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
	0x7014	32		Reserved
10.4.7	0x7018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
	0x701C	32		Reserved
10.4.9	0x7020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
10.4.10	0x7024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
10.4.11	0x702C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
10.4.12	0x7034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
10.4.13	0x703C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
10.4.14	0x7040	32	G2PTOCNT	G2P Timeout Count Register (Retry Timeout Value, TRDY Timeout Value)
10.4.15	0x7080	32	G2PSTATUS	G2P Status Register
10.4.16	0x7084	32	G2PMASK	G2P Interrupt Mask Register
	0x7088	32		Reserved
10.4.18	0x708C	32	PCIMASK	PCI Status Interrupt Mask Register

Table 16.3.1 PCI Controller Control Registers (2/2)

Reference	Address	Size	Mnemonic	Register Name
10.4.19	0x7090	32	P2GCFG	P2G Configuration Register
10.4.20	0x7094	32	P2GSTATUS	P2G Status Register
10.4.21	0x7098	32	P2GMASK	P2G Interrupt Mask Register
10.4.22	0x709C	32	P2GCCMD	P2G Current Command Register
10.4.23	0x7100	32	PBAREQPORT	PCI Bus Arbiter Request Report Register
10.4.24	0x7104	32	PBACFG	PCI Bus Arbiter Configuration Register
10.4.25	0x7108	32	PBASTATUS	PCI Bus Arbiter Status Register
10.4.26	0x710C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
10.4.27	0x7110	32	PBABM	PCI Bus Arbiter Broken Master Register
10.4.28	0x7114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for diagnostics)
10.4.29	0x7118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for diagnostics)
10.4.30	0x711C	32	PBACSTATE	PCI Bus Arbiter Current State Register (for diagnostics)
10.4.31	0x7120	64	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
10.4.32	0x7128	64	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
10.4.33	0x7130	64	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
10.4.34	0x7138	64	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
10.4.35	0x7140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
10.4.36	0x7144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
10.4.37	0x7148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
10.4.38	0x714C	32	G2PIOMASK	G2P I/O Space Address Mask Register
10.4.39	0x7150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
10.4.40	0x7158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
10.4.41	0x7160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
10.4.42	0x7168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
10.4.43	0x7170	32	PCICCFG	PCI Controller Configuration Register
10.4.44	0x7174	32	PCICSTATUS	PCI Controller Status Register
10.4.45	0x7178	32	PCICMASK	PCI Controller Interrupt Mask Register
10.4.46	0x7180	64	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
10.4.47	0x7188	64	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
10.4.48	0x7190	64	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
10.4.49	0x7198	64	P2GIOGBASE	P2G I/O Space G-Bus Base Address Register
10.4.50	0x71A0	32	G2PCFGADRS	G2P Configuration Address Register
10.4.51	0x71A4	32	G2PCFGDATA	G2P Configuration Data Register
10.4.52	0x71C8	32	G2PINTACK	G2P Interrupt Acknowledge Data Register
10.4.53	0x71CC	32	G2PSPC	G2P Special Cycle Data Register
	0x71D0-7228	-		Reserved

16.3.1.2 Interrupt signaling

PCI Controller 1 signals the following interrupt to the Interrupt Controller (IRC):

PCIC1 Interrupt (Interrupt Number: 26, PCIC1INT)

This interrupt is similar to a combination of four PCI Controller 0 interrupt signals into a single interrupt signal. For the content of each interrupt, see subsection 10.3.11.

The Ethernet Controller signals the following interrupts to the Interrupt Controller (IRC):

Ethernet Channel 0 Interrupt (Interrupt Number: 6)

Ethernet Channel 1 Interrupt (Interrupt Number: 5)

For the content of these interrupts, see 16.4.3.9 Interrupt Source Register.

16.3.1.3 IDSEL signal connections

Internal PCI Bus AD (Address, Data) signals are connected to Ethernet Controller 0, 1 IDSEL signals. Each connection destination follows below.

Ethernet Channel 0 IDSEL signal: Internal PCI Bus AD[31]

Ethernet Channel 1 IDSEL signal: Internal PCI Bus AD[30]

16.3.1.4 On-chip PCI bus clock frequency

The on-chip PCI bus can be programmed to operate at one half or one fourth of the GBUSCLK frequency. Apply a logic High to ADD[16] at boot time to set the on-chip PCI bus clock frequency to one fourth of GBUSCLK. When it is necessary to increase the on-chip PCI bus clock frequency to one half of GBUSCLK, program the relevant register bits in the following order:

1. Set the PCIC1RST bit in the Clock Control (CLKCTR) register to 1. This asserts the reset signals to the PCIC1, ETHER0 and ETHER1 modules.
2. Set PCI1-66 bit in the Chip Configuration (CCFG) register to 1.
3. Set the PCI1DMD bit in the Chip Configuration (CCFG) register to 1.
4. Clear the PCIC1RST bit in the Clock Control (CLKCTR) register to 0. This asserts the reset signals to the PCIC1, ETHER0 and ETHER1 modules.

16.3.2 Data structure

The Ethernet Controller exchanges control information and data using the following data structure.

- Frame Descriptors
- Buffer Descriptors
- Data Buffer

Figure 16.3.1 shows the relationships of the above data structures.

A frame descriptor consists of a 4-byte pointer that points to the next frame, a field dedicated to a system or application program, a frame status field, a control field for all frames, and a corresponding group buffer descriptor. The buffer descriptor contains a pointer to the data buffer and buffer control information. The data buffer is a data storage area in Byte units that is stored in either the Little Endian or Big Endian order. For details on these data structures, see 16.3.7 Memory configuration.

The DMA engine responds to the transfer of data that is positioned at the byte boundaries. Frame descriptors must be aligned to 16-Byte boundaries. Also, buffer descriptors must be aligned to 8-Byte boundaries. The DMA engine Burst transfers 4-Byte aligned data as much as possible. However, whether data buffers are aligned to 4-Byte boundaries or not does not have much effect on performance. The DMA engine accesses only words or parts of words not aligned to 4-Byte boundaries at the start and end of Block transfer. Part way through Block transfer however, the DMA engine performs word-unit 4-Byte access.

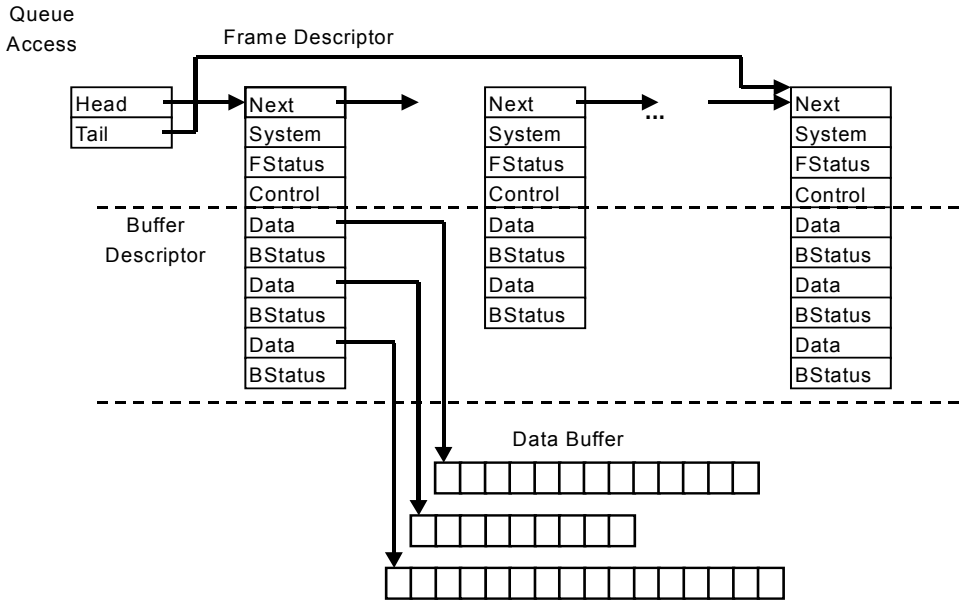


Figure 16.3.1 Data Structure Outline

16.3.3 System control model

The Ethernet Controller as two programmable modes: the Interrupt Drive mode and the Polling mode.

- Interrupt Drive mode
  This mode generates an interrupt each time a packet or group packet transmits or receives.
- Polling mode
  This mode always polls to see if there is a subsequent packet to transmit or receive. When data or a descriptor is added to the frame descriptor queue, the Ethernet Controller performs transmission. The Host CPU checks the status of the frame descriptor for the received packet and judges reception completion. The CPU does not use an interrupt.

By enabling or disabling interrupts for each frame descriptor, the Ethernet Controller can arrange the stage layout to enable the handling of multiple packets between interrupts. In this way, the ability of the system to handle multiple packets at once and to cache the handling instructions makes it possible to reduce interrupt handling overhead and improve performance.

Furthermore, unless major errors occur, you can set the Ethernet Controller to not issue interrupts. In this situation, you have to put several controls in the Enable state so start/end of traffic and polling in the Idle state is handled efficiently. For details, see 16.3.9 DMA Operation.

16.3.4 Functional overview

Figure 16.2.1 is a diagram of the entire Ethernet system. Figure 16.3.2 and Figure 16.3.3 are more detailed function block diagrams of the Ethernet Controller. Figure 16.3.2 is a function block diagram of DMA. Figure 16.3.3 is a function block diagram of MAC.

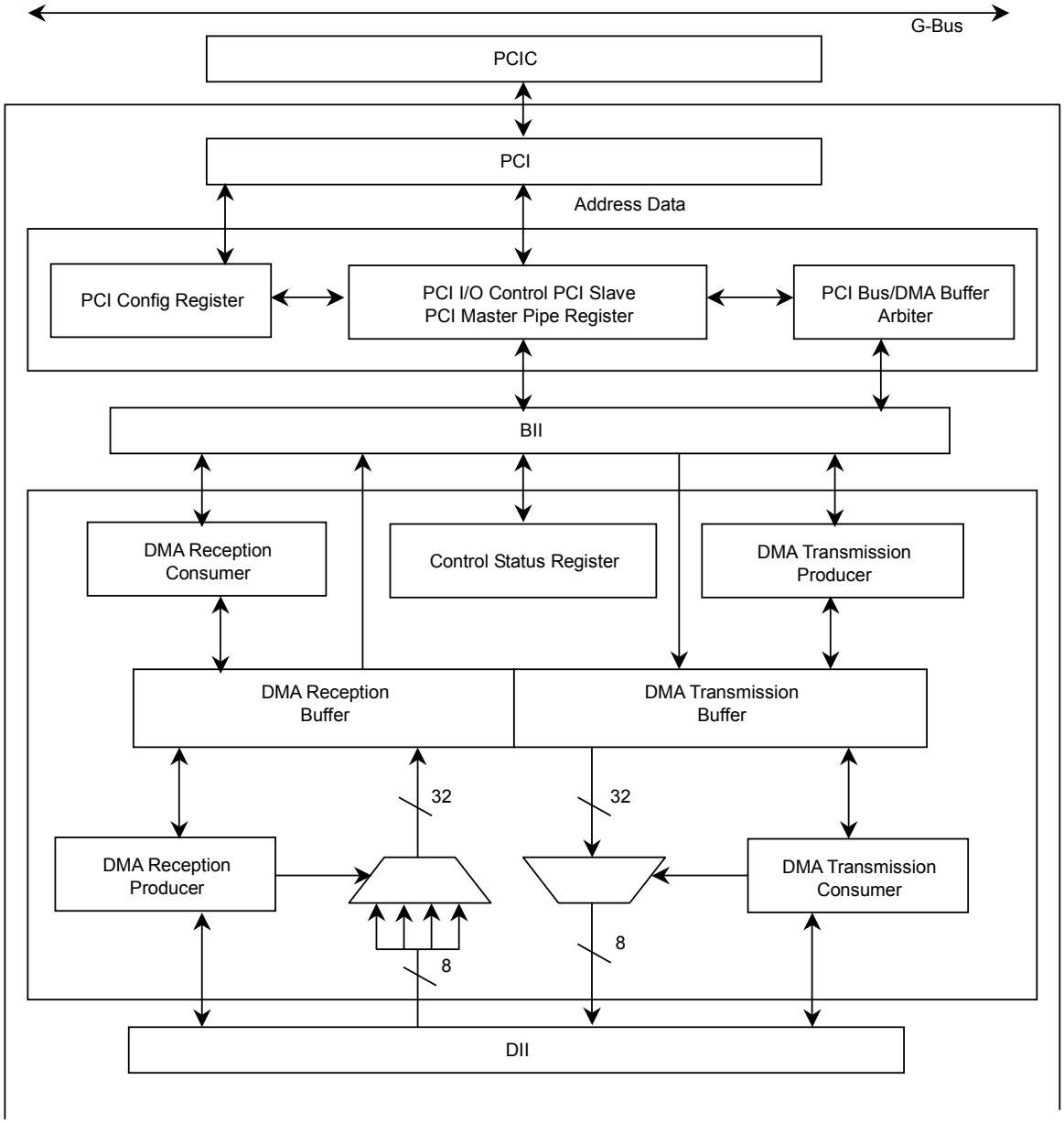


Figure 16.3.2 DMA Function Block

Figure 16.3.2 and Figure 16.3.3 both show the relationship with the DMA-independent interface (DII). DII is an internal interface that has simple handshake signals and interfaces MAC-internal FIFO with the DMA block-internal DMA buffer. The two FIFOs in the MAC are 1 Byte wide. The DMA block contains DMA buffers that are wider and deeper than MAC. Figure 16.3.2 shows the relationship with the bus-independent interface (BII). The BII is an internal interface that has Bus Request/Bus Grant protocols for Burst access performed by the Bus Master. The BII makes it possible to connect the DMA engine to a 32-bit PCI Bus Controller or another bus controller.

16.3.4.1 Overview of PCI and DMA

Figure 16.3.2 shows the PCI Bus configuration, blocks that perform control, DMA buffer and data exchange.

- PCI I/O Control Block
  Generates and acknowledges PCI control signals.
- PCI Slave Control Block
  Acknowledges and controls transactions when the Ethernet Controller is the target device.
- PCI Master Control Block
  Starts and controls transactions requested by the DMA engine of the Ethernet Controller.
- PCI Pipe Register Group
  Buffers data so the DMA engine can continue Burst transfer with 1-1-1-1 timing over a long period of time.
- PCI Configuration Register
  Flexibly performs PCI system setup.
- DMA Control Status Register
  Sets up and controls DMA.
- Arbiter Block
  Arbitrates access to the PCI Bus or the DMA buffer.
- Producer Block, Consumer Block
  Controls transmission/reception data output/input to/from the DMA buffer.

The DMA buffer the DMA engine controls has a transmission area and a reception area. The DMA Transmission buffer stores the data and status information of multiple packets that are currently being transmitted. The DMA Reception buffer stores the data and status information of multiple packets that were received. Each buffer has a Producer Block that controls the data stored in a buffer and a Consumer Block that controls data removed from a buffer.

The PCI Bus/DMA Buffer arbiter determines whether the consumer or producer State Machine has higher priority access permission to the PCI Bus/DMA Buffer. The priority changes dynamically and, if possible during Burst transfer, gives priority to the State Machine that controls the PCI Bus. However, when either the MAC Reception FIFO becomes nearly full or the MAC Transmission FIFO becomes nearly empty during transmission, priority is given to that FIFO. In other situations, the round-robin scheme is used to provide service in a fair manner.

The DMA Controller block consists of the circuits required for the Bus Master to perform read/write operations via the PCI Bus.

- Busrt Size Control Circuit
  Optimizes PCI and system performance.
- Transmission Threshold Control Circuit
  Matches transmission latency to PCI Bus latency.
- Big Endian Byte Translation Circuit
  Supports data transfer to Big Endian format processors.
- Buffer Distribution/Sharing Control Circuit
  Can distribute then store one frame in multiple buffers. Can also store multiple frames in a single buffer and increase the usage efficiency.
- Polling Control Circuit
  Polls transmission packets. Is optional.
- Transmission Wakeup Control Circuit
  Performs control required to start transmission when the data is prepared.
- Early Notification Circuit
  Circuit that generates signals for starting to handle reception data before the data ends.
- Interrupt Enable Control Circuit
  Adjusts controller operation as the protocol requires.

16.3.4.2 MAC overview

Figure 16.3.3 is a function block diagram of MAC. MAC consists of a Transmission Block, Reception Block, Control/Status Register group, Flow Control Block, and a Serial Controller. The Serial Controller is an MII station management interface.

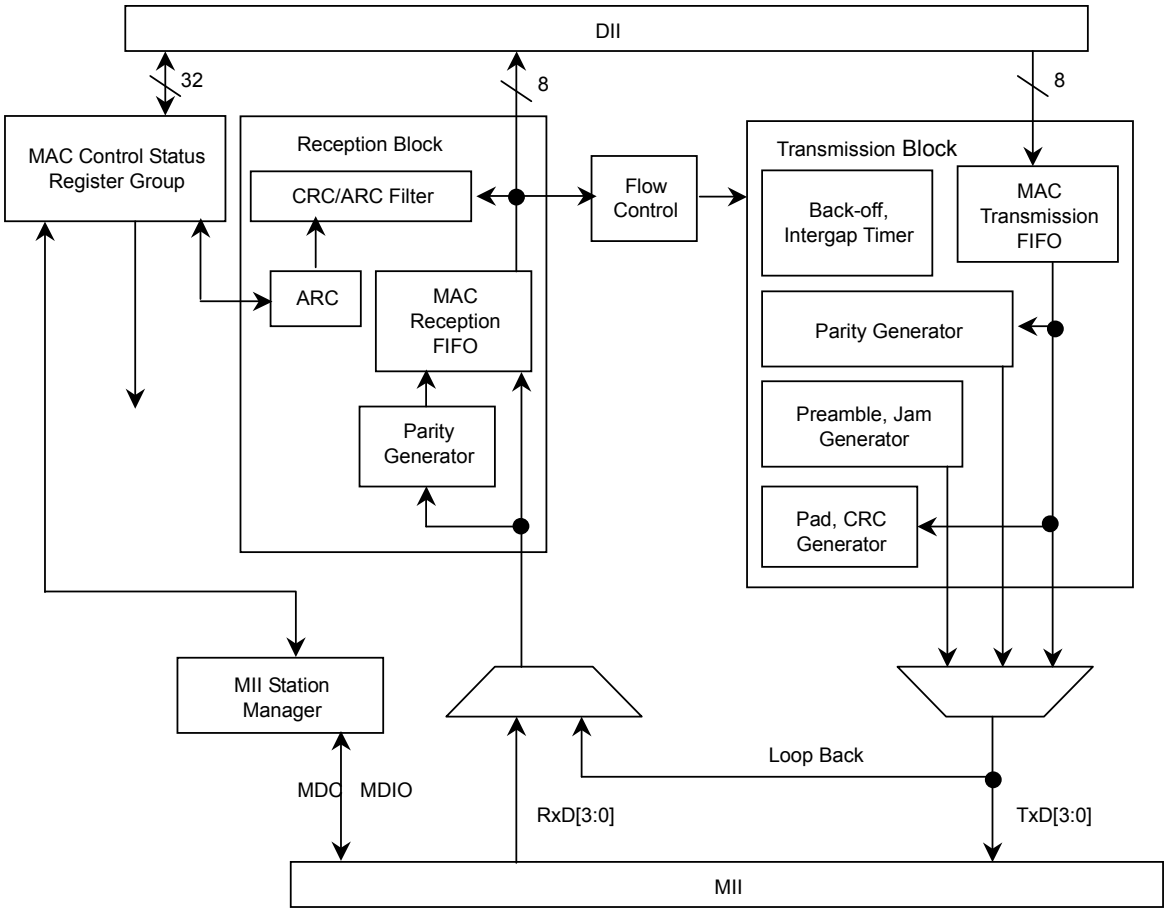


Figure 16.3.3 MAC Function Block

The media-independent interface (MII) transfers signals between the 100Base-T compatible physical layer and the Transmission/Reception Block. MII is described in the IEEE 802.3 standard.

The Transmission Block buffers data transmitted from the MAC Transmission FIFO, assembles packets, and then transfers them to the MII. Included in the Transmission Block are circuits that generate Preamble Bytes, Jam Bytes, Pad Bytes, and CRC values. Also included is a parity check circuit, a timer that creates back-off delay when a collision occurs, and a time that creates a cap between packets that were transmitted.

The Reception Block expands packets received from MII, and then stores them in MAC Reception FIFO. Included in the Reception Block are a circuit that checks the CRC value, a circuit that generates parity to protect data in the FIFO, and a circuit that checks the packet length. Also included is an address recognition circuit (ARC) that, based on the receiving address, judges whether to accept or discard a packet.

You can use a loop back circuit to separate MAC from the MII and physical layer and perform MAC layer testing.

The MAC function block performs control to perform network manipulation such as the following:

- Control to include a stop request at the end of a packet currently being transmitted or received, and enable or disable the transmission/reception circuit
- Interrupt enable/disable control for each condition
- Address recognition control for up to 21 addresses
- Counter, status bit for collecting network management data
- Loop back or other control that aids network failure diagnosis
- Pause operation enable control for pausing the transmitter when a MAC control frame described by Pause operation is received
- MAC control frame transmission control for enabling Pause or other MAC control frame generation even when the transmitter has paused
- MAC control frame pass through control for enabling hardware or software to handle MAC control frames in another format

16.3.5 DMA function block

16.3.5.1 DMA Transmission Controller

The DMA Transmission Controller consists of two State Machines: Producer and Consumer. The Producer checks the frame descriptor of the transmission queue, then controls data transfer from the transmission queue to the DMA Transmission buffer. Also, after transmission ends, the Producer controls the writing of transmission status information that expresses the MAC status information for the transmitted data. The Consumer controls transfer of data from the MAC transmission engine or a large-capacity DMA transmission buffer to a small MAC transmission FIFO.

16.3.5.2 DMA Reception Controller

The DMA Reception Controller consists of two State Machines: Producer and Consumer. Producer controls data transfer from MAC Reception FIFO to the DMA reception buffer. Consumer allocates buffers from the empty buffer list, writes the frame descriptor and related buffer descriptor of the packet in the free descriptor area, then controls data transfer via the PCI Bus from the DMA Reception buffer to the system memory.

16.3.6 MAC function blocks

16.3.6.1 MAC Transmission Block

The Transmission Block takes charge of data transmission. The Transmission Block is compliant with the IEEE 802.3 carrier sense multiple access with collision detection method (CSMA/CD) protocol. This block also supports full duplex modes that can simultaneously perform transmission and reception. The Transmission Block consists of the following parts.

- Transmission FIFO, FIFO Control Counter
- Preamble, Jam Oscillator
- Pad Byte, CRC Generator
- Parity Checker
- Back Off, Intergap Timer

For details of the Transmission Block, see 16.3.8 MAC operation.

16.3.6.2 MAC Reception Block

The Reception Block takes charge of data reception. The Reception Block is compliant with the IEEE 802.3 carrier sense multiple access with collision detection method (CSMA/CD) protocol. This block also supports full duplex modes that can simultaneously perform transmission and reception. The Reception Block consists of the following parts.

- Reception FIFO, FIFO Control Counter
- Address Recognition ARC Block
- CRC Generator, Tester
- Parity Generator

For details on the Reception block, see 16.3.8 MAC operation.

16.3.6.3 Flow Control Block

The Flow Control Block has the following functions:

- Recognize MAC control frames the Reception Block received
- Transmit MAC control frames (even when the transmitter is paused)
- Pause operation timer and counter
- Command/Status Register Interface
- Options for handing off MAC control frames to the software driver

The reception circuit in the Flow Control Block recognizes the MAC control frame, and then performs the following Pause operation. First, the Data Length/Data Type field must have a specific value for a MAC control frame. Second, the ARC must recognize the destination address. Third, the frame length must by 64 Bytes including CRC. Fourth, the CRC test result must confirm the correctness of the frame. Finally, the frame must include a valid Pause operation code and operand.

If the Data Length/Data Type field does not contain a specific value that expresses a MAC control frame, MAC performs no operation and the packet is treated as a typical packet. If the ARC cannot recognize the destination address, MAC ignores the packet. If the packet length including CRC is not 64 Bytes, MAC does not perform Pause operation. In this case, if Pass Through is in the Enabled state, the driver is handed off.

If the Control bit of the Transmission Status Register is set, you can detect full duplex mode Pause operation or other MAC control function frames even if the transmitter itself is in the Pause state.

The Flow Control Block has two timers for Pause operation and a Control/Status Register for each timer. One timer, register pair is used when the received packet pauses the transmitter. The other timer, register pair is used to estimate the pause status of the transmission destination after the transmitter transmits a Pause command. The Command/Status Register Interface is used to access the Transmission Control Register, the Transmission Status Register, the Reception Control Register, and the Reception Status Register. You can use these registers to start transmission of a MAC Control frame, enable or disable the MAC control function, or access the Flow Control counter.

You can use the Control bit to select whether to completely handle MAC Control frames inside the Ethernet Controller or hand them off to the software driver. Therefore, default flow control is possible even if the software driver itself does not support flow control.

16.3.6.4 MAC Control Register, Status Registers

MAC has a group of Control Registers and Status Registers. These registers are used to control the Transmission Block or Reception Block, display MAC status, are used in the communication interface with CAM, and are used in the interface with the MII Station Manager. These registers can also access from PCI using the memory map or I/O map. For details on the MAC Control Register or Status Register, see 16.4.5 MAC Control, Status Register group.

16.3.6.5 MII Station manager

MAC in the Ethernet Controller handles station management data signals (MDIO and MDC) from the MII Controller, but it does not interpret them. Using a serial interface defined by MII, the MII Station Manager reads and writes to/from the Control Registers and Status Registers in a PHY device whose configuration is set up.

When it is necessary to access these registers to negotiate configuration using an application specialized for a bridge, router, switching hub, etc., you can use the MAC Control Registers and Status Registers to trigger reads and writes through the Station Management Data Interface. For details, see 16.4.5.5 Station Management Registers.

16.3.6.6 Reception packet alignment

When storing Reception packets in the buffer, the Ethernet Controller can skip by the set byte count. By default, the Ethernet Controller does not skip Reception packets and these packets are on double-word boundaries. You can set the DMA Control Register to have the first buffer skip 1-3 Bytes. This function is convenient when aligning packets when internally necessary by handling IPs and decoding.

16.3.7 Memory configuration

This subsection describes the data structure the PCI-mounted Ethernet Controller uses when exchanging data with the Host system. The data structures are stored in system memory. There are three basic data structures:

- Frame descriptor
- Buffer descriptor
- Data buffer

These data structures are used in the following manners:

- Transmission queue: Listing the frame descriptors of packets ready for transmission
- Reception queue: Listing the frame descriptors of received packets
- Buffer list: Listing the frame descriptors that have unused buffers for receiving data

Figure 16.3.1 shows an outline of each data structure. This subsection describes each data structure in detail. Depending on the type of data structure, some queues in use may include different information. Each following item will describes this in detail. During continuous polling operation, a queue never becomes empty once it is created. One frame descriptor is always appended to the end of a queue. This dummy frame descriptor is used in processes that generate a descriptor to be transmitted. For details, see 16.3.9 DMA Operation. To start transmission, the system sets the address of the first frame descriptor in the Transmission queue in the Transmission Frame Pointer Register. The Ethernet Controller transmits while tracing the Transmission queue, then updates the status of the transmitted packet. The owner bit of the Frame Descriptor Status Field and Frame Descriptor Control field indicates "Transmission complete". This enables the system software to handle the queue in situations such as when releasing the buffer. The Ethernet Controller fetches the buffer from the buffer list, and then writes the new frame descriptor or new buffer descriptor in a free descriptor area. Subsection 16.3.9 DMA Operation. describes this.

16.3.7.1 Frame descriptor

A frame descriptor consists of a pointer to the next frame descriptor in the queue, a System Data field, a Frame Length field, a Control field, and a Status field.

Table 16.3.2 shows the frame descriptor format.

Table 16.3.2 Frame Descriptor Format

Byte 3	Byte 2	Byte 1	Byte 0	Offset
FDNext				00 h
FDSystem				04 h
FDStat				08 h
FDCtl		FDLength		0C h

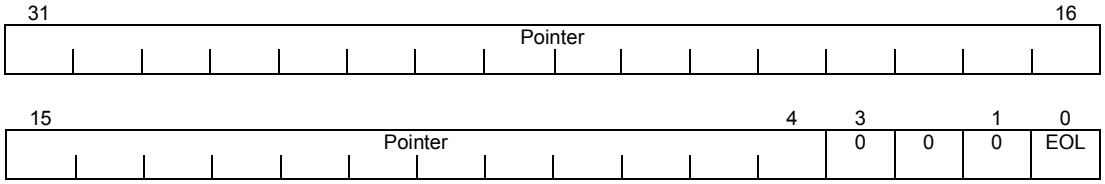
FDNext	Next frame descriptor	Address of the next frame descriptor in the queue
FDSystem	Frame system data	Is used by the system or application software
FDStat	Frame descriptor status	Status field of this frame descriptor
FDCtl	Frame descriptor control	Control field of this frame descriptor
FDLength	Frame descriptor length	Field that expresses the length of this field

The Ethernet Controller retains the Frame System Data field (FDSystem). The FDSystem field can be used by system or application programs. The initial value of the frame descriptor written to the Reception queue is fetched from the frame descriptor of the current buffer list.

Depending on the queue type, the usage of the FDNext, FDCtl, FDStat, or FDLength fields varies. See the following items for an explanation.

16.3.7.1.1 FDNext field (next frame descriptor)

The FDNext field stores a flag that indicates the end of list (EOL) or a pointer to the next frame descriptor in the same queue. The frame descriptor must be aligned to a 16-byte boundary. In other words, bits 0-3 of the valid pointer must be "0".



Bit(s)	Mnemonic	Field Name	Description
31:4	Pointer	28-bit Pointer	28 bit Pointer When EOL=0, the upper 28 bits of the address of the next frame descriptor in this queue is stored.
3:1			Is fixed to 000
0	EOL	End Of List flag	EOL 0: Pointer is valid. 1: End of list. You have to wait until this flag is cleared.

Figure 16.3.4 FDNext Field

Regardless of the queue type, by setting the EOL bit, you can use the FDNext field stop the list consumer (consumption side processing mechanism). The consumer must wait for the list producer (generation side processing mechanism) to clear the EOL bit and store a valid pointer. In the case of a buffer list queue, you can use the FDNext field to create a chain to the next buffer pool from any buffer pool. When not creating a buffer list chain, the software driver has to set its own address in the FDNext field. Doing so makes it possible for the Ethernet Controller to recheck so it can reuse the same buffer area. You can also set the EOL bit and stop the Ethernet Controller.

16.3.7.1.2 FDSystem field (frame descriptor system data)

The FDSystem field is a 32-bit field that is reserved for use by the system software. You can use this field to store pointers to the information table, pointers to C++ virtual functions, etc.

The Transmission queue does not use the FDSystem field.

In the case of a Reception queue, the Ethernet Controller copies the content of the FDSystem field of a frame descriptor that retains a buffer list to which is allocated the first buffer descriptor that stores the frame.

16.3.7.1.3 FDStat field (frame descriptor status)

The Transmission queue or Reception queue uses the FDStat field to display the transmission/reception complete status. For the meaning of each Status bit, see the descriptions of the TX_Stat Register and Rx_Stat Register in subsection 16.4.5 MAC Control, Status Register group.

The reception empty buffer list does not use the FDStat field.

16.3.7.1.4 FDLength field (frame descriptor length)

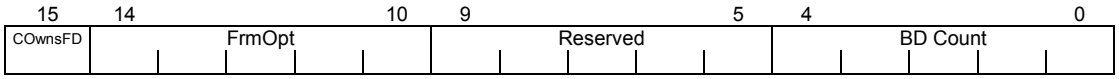
The Transmission queue does not use the FDLength field.

In the case of the Reception queue, the Ethernet Controller sets the FDLength field to the total packet length.

In the case of a buffer list, the FDLength field is used to count the number of empty buffer descriptors allocated to the queue. The Ethernet Controller accesses the buffer list frame descriptor through the Buffer List Frame Pointer Register. When the Ethernet Controller encounters a buffer that it does not own, it sets the BL_Ex bit of the Interrupt Source Register and then waits for the system to clear it. The Ethernet Controller treats the FDLength field value as the upper limit, then reads the buffer descriptor. As the Ethernet Controller nears the end of the list, it reads out the next frame descriptor that the FDNext field indicates, as described in 16.4.3.8 Free Descriptor Area (FDA) Register.

16.3.7.1.5 FDCtl field (frame descriptor control)

This item shows the FDCtl field configuration and application. The Transmission queue or Reception queue uses the COwnsFD bit to synchronize the Ethernet Controller and System process. The Transmission queue or buffer list uses the FrmOpt field. The Transmission queue or Reception queue uses the BDCount field (the buffer list uses the Frame Descriptor Length field (FDLength) to enable the use of buffer pools larger than the BDCount field permits). For each reception packet, an excessive buffer descriptor error is issued when more than 28 buffer descriptors are used.



Bit(s)	Mnemonic	Field Name	Description
15	COwnsFD	Frame Descriptor Owner	COwnsFD 1: The Ethernet Controller owns the frame descriptor after the system sets the COwnsFD bit. 0: The system owns the frame descriptor after the Ethernet Controller clears the COwnsFD bit.
14:10	FrmOpt	Frame Option	Frame Option Control option for each frame (see the following description)
4:0	BDCount	Buffer Descriptor Count	BDCount This is the number of allocated buffer descriptors (1-28).

Figure 16.3.5 FDCtl Field

The Transmission queue uses the FrmOpt field and sets the transmission characteristics of each packet.

- 10000 Big Endian order
- 01000 Issue interrupt after transmission
- 00100 Do not add CRC
- 00010 For short frames, do not add PAD

You can use combinations of the above bits to set various transmission characteristics. For example, "01110" means "Little Endian, issue interrupt after transmission, no CRC, no padding for short packets".

In applications such as hubs where transmission packets are being received from both Big Endian and Little Endian transmission sources, controlling Big Endian for each packet is useful. In computer applications, it is easier to use the global Big Endian control bit described in 16.4.3.1 DMA Control Register.

Unless you set a global enable bit like that described in 16.4.3.6 Reception Fragment Size Register, the Reception buffer list uses the option set in the FrmOpt field to control packing and the endian.

- 10000 Big Endian order
- 00001 Enables packing for this frame's buffer, ignores any global enable bit

When packing is enabled, the RxFragSize Register controls the packing algorithm.

16.3.7.2 Buffer descriptor

Each buffer descriptor consists of a pointer to the data buffer, a control byte, a status byte, and a 2-byte Buffer Length field. Table 16.3.3 indicates the buffer descriptor format.

Table 16.3.3 Buffer Descriptor Format

Byte 3	Byte 2	Byte 1	Byte 0	Offset
BuffData				00h
BDCtl	BDStat	BuffLength		04h

BuffData	Buffer Data Pointer	Address of the buffer that stores data (32 bits)
BDCtl	Buffer Descriptor Control	Control of this buffer descriptor
BDStat	Buffer Descriptor Status	Status of this buffer descriptor
Buff Length	Buffer Length	Field that expresses the length of this buffer

When a buffer is listed in the buffer list, the BuffData field indicates the buffer head and the BuffLength field indicates the size of buffers allocated as unused buffers. When the buffer is used in the Transmission queue or Reception queue, the BuffData field indicates the descriptor head and the BuffLength field indicates the data length.

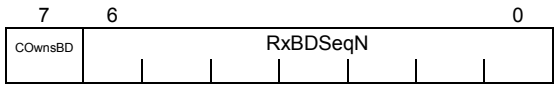
When a buffer is placed in the buffer list, the system software is responsible for allocating the Buffer Length field and setting the size.

Similar to the frame descriptor, usage of the BDCtl field or BDStat field varies depending on the queue type.

Note: As shown in step 4 of 16.3.8.4 Frame transmission procedure, more than 8 bytes of data is required as a Transmission packet data. One Transmission packet cannot consist of multiple data buffers, so you have to implement your design so the BuffLength (buffer length) total is 8 bytes or more.

16.3.7.2.1 BDCtl field (buffer descriptor control)

The transmission queue does not use the BDCtl field. In the case of the reception queue, the number of buffer descriptors per buffer area is set in the BDCtl field. The first buffer descriptor in a frame is number 0, and the subsequent buffer descriptors are number 1 and number 2.



Bit(s)	Mnemonic	Field Name	Description
7	COwnsBD	Buffer Descriptor Owner	COwnsBD 1: The Ethernet Controller owns the buffer descriptor. When the system sets the COwnsBD bit, the buffer can freely receive data. 0: The system owns the buffer descriptor. When the Ethernet Controller clears the COwnsBD bit, it indicates that the buffer is full.
6:0	RxBDSeqN	Reception Buffer Descriptor Number	RxBDSeqN After receiving data, this field indicates the running number of this buffer in the current buffer area.

Figure 16.3.6 BDCtl Field

In an empty buffer list, the BDCtl field is used to record buffer ownership. This makes it possible to allocate the buffer descriptor and synchronize release of the buffer. This also prevents the Ethernet Controller from using the buffer in a loop before the system empties the buffer.

16.3.7.2.2 BDStat field (buffer descriptor status)

In the case of a transmission queue, the Ethernet Controller does not use the BDStat field. In the case of a reception queue, the BDStat field is used as the buffer ID. This value is copied from an empty buffer queue.

Bits	Mnemonic	Field Name	Description
7:0	RXBDID	Reception Buffer Descriptor ID	RXBDID Value of the buffer descriptor identification number

Figure 16.3.7 BDStat Field

In the case of an empty buffer queue, the BDStat field is used to hand the buffer descriptor identification number to the Ethernet Controller.

Note: You can only apply buffer descriptor identification numbers freely if there are no more than 256 buffers in a single buffer pool.

16.3.8 MAC operation

This subsection describes MAC operation in detail with respect to the following content:

- Format of MAC frames and packets
- Initializing MAC
- Access MAC Registers
- Frame transmission procedure
- Frame reception procedure
- Operation of the address recognition circuit (ARC)
- Pause operation in full duplex transfers
- Signaling transmission/reception errors
- Accessing station management data

16.3.8.1 Format of MAC frames and packets

Figure 16.3.8 shows the format of IEEE 802.3 Ethernet packets. Standard packets have the following fields.

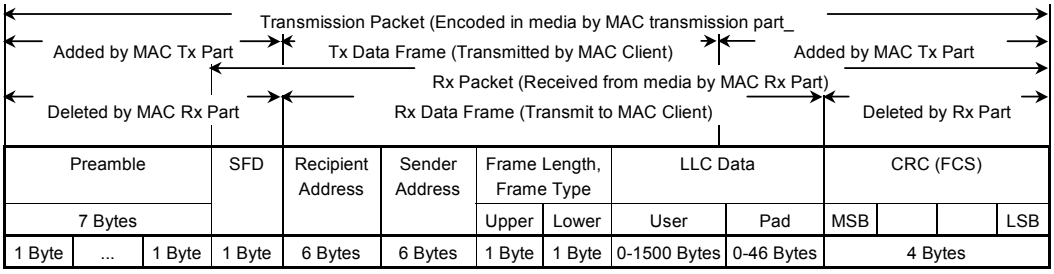


Figure 16.3.8 Ethernet Packet Frame Fields

- Preamble— repeat the same byte 7 times
  Each byte is fixed to 10101010 and is transmitted from left to right. The preamble is sometimes shortened during transmission and is not necessary when receiving data.
- Start Frame Delimiter (SFD)— 1 Byte
  The SFD is fixed to 10101011 and is transmitted from left to right. It is necessary when receiving data.
- Recipient address— 6 Bytes
  This is a unicast or multicast address. ARC can use the recipient address and can freely perform address filtering.
- Sender address— 6 Bytes
  MAC does not check the sender address. However, to make the sender address a valid station address, the first bit (LSB) of the first Byte transmitted must be “0”.
- Frame length or frame type— 2 Bytes
  Is transmitted upper byte first. In the IEEE802.3 standard, a value of 1500 or less is defined as the frame length, and any value greater than 1535 is defined as the frame type. If the value is 1500 or less, it indicates the byte count of the logic link control (LLC) data in the Data field. MAC recognizes 8808h as the MAC control frame and specially recognizes 8100h as a tagged VLAN frame.

- Logic link control (LLC) data— 0-1500 Bytes
  LLC data consists of two fields: User Data and Pad Data.
- User Data— 0-1500 Bytes
- Pad Data— 0-46 Bytes
  When the user data is less than 46 Bytes long, MAC can add padding bytes and make the LLC data 46 Bytes long.
- Cyclic redundancy check (CRC)— 4 Bytes
  Also referred to as the frame check sequence (FCS), this value is calculated from all other fields except the preamble, SFD, and CRC itself.

The preamble, SFD, pad data, and CRC are added on by the transmission side. You can also pad data using the software. CRC can use the Transmission Control Register to suppress padding. You can use the Reception Control Register to control deletion of CRC. You can delete pad data using either the DMA engine or the software driver.

Except for CRC, the MAC transmits each byte starting from the least significant byte. In this document, all bytes transmitted or received are collectively referred to as “packets”. The term “frame” refers to a part that you provide during transmission or a part that is provided to you during reception.

Standard IEEE802.3 frames are transformed by various factors and options.

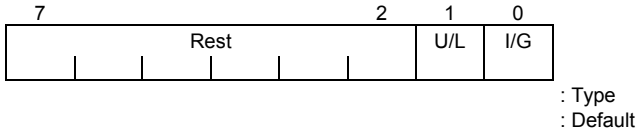
- Depending on the PHY, there are cases where the preamble length is not 7 Bytes. Sometimes a repeater shortens the preamble. The transmission part sends the preamble in a standard format, but the reception part does not care even if there is no preamble or the preamble is more than 7 Bytes long. SFD follows the preamble.
- When in the Short Packet mode, LLC data less than 46 Bytes long is allowed. In this case, there are options for suppressing padding when transmitting data or for allowing the reception of short packets.
- When in the Long Packet mode, LLC data larger than 1500 Bytes is allowed. There is an option for allowing the reception of long packets.
- There is an option for suppressing the addition of CRC fields.
- There is also an option for allowing the reception of packets that do not have a valid CRC field.

16.3.8.1.1 Format of recipient addresses

Bit 0 of a recipient address specifies the address type. This bit indicates whether the address is an individual address or a group address. Group addresses are also referred to as multicast addresses. Individual addresses are also referred to as unicast addresses. Broadcast addresses are special group addresses that are FF-FF-FF-FF-FF-FF hexadecimal addresses.

Bit 1 distinguishes locally managed addresses from globally managed addresses. If an address is managed globally (universal), bit 1 is set to "0". If an address is allocated locally, bit 1 is set to "1". Bit 1 is set to "1" for broadcast addresses.

Recipient address (first Byte)



Bit(s)	Mnemonic	Field Name	Description
7:2	Rest	Remaining Bit	Rest Remaining bits of the first Byte are the recipient bits.
1	U/L	Universal/Local	U/L 0: Universal address 1: Local address
0	I/G	Individual/Group	I/G 0: Individual address 1: Group address

Figure 16.3.9 Format of Recipient Addresses

16.3.8.1.2 Special flow control recipient address

The IEEE802.3 standard specially prescribes a recipient address of 01-80-C2-01-00-01 for PAUSE manipulation packets to realize full duplex flow control. In order for the MAC to receive packets including the special recipient address for PAUSE manipulation, you have to set the address in one entry in ARC memory, enable that entry, and then activate ARC itself.

The details are described in the following paragraphs. However, part of the ARC entry uses the SdPause bit of the Transmission Control Register and is also used when generating flow control frames.

16.3.8.2 Initializing MAC

After powering up or after performing reset, the MAC Control Status Register is initialized as described in section 16.4 Registers.

The transmission collision count and ARC data is not initialized when you power up or reset the Ethernet Controller. The transmission collision count is reset when a new packet is transmitted. Initialize ARC memory before enable ARC.

16.3.8.3 MAC Register access

Access to the MAC Register is controlled through the PCI Bus Interface. For details on read access and write access of the register, including the MAC Control Register, see 16.3.9 DMA operation.

16.3.8.3.1 Clearing special registers

The Lost Packet Error Count Register is cleared when it is read. This makes it possible to synchronize it to software drivers that tabulate the total error count.

The Transmission Status Register and Reception Status Register are cleared at the beginning of the next packet. Therefore, values read from the Register Interface may not be stable. These register values are stored in the FDStatus field of the frame descriptor in memory for each transmission or reception packet. You have to use the software to check the status value retained in the system data structure.

16.3.8.4 Frame transmission procedure

(1) To transmit a frame, the Transmission Enable bit (TxEn) of the Transmission Control Register must be set, and the Transmission Halt Request bit (TxHalt) must be cleared. Furthermore, the Halt Transmission Immediately bit (TxHalt) and the Halt Request bit (HaltReq) of the MAC Control Register must be cleared. Usually, the above conditions (such as storing a valid frame descriptor address in the Transmission Frame Descriptor address) are set after initializing the DMA Controller. In this way, MAC instructs the DMA engine to transmit a frame to MAC Transmission FIFO. At this time, the DMA Transmission Controller controls transfers to the MAC Transmission FIFO.

(2) The MAC Transmission Block starts transmitting data in the FIFO. However, the first 64 Bytes are held in the FIFO until the net is fetched. Then, the MAC Transmission Block requests the next data and continues to transmit until the DMA Transmission Controller indicates the end of transmission data. The MAC Transmission Block generates pad bytes when necessary, adds CRC to the end of a packet, then ends transmission. The MAC Transmission Block will then set the Transmission Complete bit (Comp) of the Transmission Status Register to signal that transmission has ended. Finally, depending on the Interrupt Enable Register settings, an interrupt may occur.

(3) Data transfer via the MII Interface is driven by a 25 MHz or 2.5 MHz MII Transmission Clock (Tx_clk).

(4) The MAC Transmission Block must not start transmitting to the net until 8-byte data is stored in the MAC Transmission FIFO. The first 8 bytes transmitted are the preamble and Start Frame Delimiter (SFD), so the allowable DMA latency when starting transmission is 16-byte time. The DMA Transmission Block does not transmit data to the MAC Transmission FIFO until either the entire packet is stored in the DMA Transmission buffer or the byte count set in the Transmission Threshold Register is stored in the DMA Transmission buffer. If a Transmission Underrun error is generated, you can avoid an underrun by increasing the setting of the Transmission Threshold Register.

(5) The MAC Transmission Block performs a parity check. If a Parity error is generated, the MAC Transmission Block aborts transmission, then resets the FIFO and sets the Transmission Parity Error bit (TxPar) of the Transmission Status Register.

16.3.8.4.1 IEEE 802.3 transmission protocol

The MAC Transmission Block consists of three State Machines. The Main State Machine executes MAC layer protocol and controls the other two State Machines (Gap State Machine and Back Off State Machine). The Gap State Machine fetches the timing gap between packets, then counts them. The Back Off State Machine executes the Back Off or Resend algorithms of the 802.3 CSMA/CD protocol.

16.3.8.4.2 Interpacket gap (IPG) timing

When in the half duplex mode, the Gap State Machine measures 96-bit time from the point when the Carrier Sense signal is deasserted. This time becomes the interpacket gap. Gap State Machine splits the 96-bit time into 64-bit and 32-bit time, then precisely controls the timing at which transmission starts. If there is traffic with the first 64-bit time, the Gap State Machine resets the counter to 0, then starts counting from the beginning. If there is traffic during the remaining 32-bit time, the count continues as is and signals that 96-bit time has elapsed.

When in the full duplex mode, Gap State Machine starts counting along with transmission completion, then signals transmission completion after 96-bit time passes.

16.3.8.4.3 Collision process and back off

When the Main State Machine detects a collision, it starts up the counter of the Back Off State Machine, waits for the Back Off time to elapse, then attempts to resend the packet that caused the collision. The Back Off time is a multiple of 512-bit time (including × 0). When a collision happens in the same packet, the Main State Machine advances the internal Trial Count counter by 1 each time this happens. Then, it causes an 11-bit pseudorandom number generator to generate random numbers and output a subset of it. Each time this is retried, the subset increases in size by 1 bit. In this way, the following formula is invoked by the hardware.

$0 \leq r < 2^k$

$k = min(n, 10)$

r is the slot time count that the MAC must wait when a collision occurs. *n* is the number of retransmission attempts. For example, after the first collision, n=1 and *r* is a random number between 0 and 1. In this case, the pseudorandom number generator is 1-bit wide and generates either 0 or 1 as a random number. For the second and subsequent attempts, *r* is a random number between 0 and 3. In other words, State Machine is n=2, so the pseudorandom number generator looks at the lower 2 bits and generates random numbers between 0 and 3.

To reduce the statistical probability of similar random number sequences being generated between multiple MACs using the same random number generator, MAC uses the CRC value of the packet that was previously transmitted successfully to recalculate a basic random number sequence.

16.3.8.4.4 MII transmission operation

If there is data to be transmitted, there is no problem with the interpacket gap, and MII preparations are complete (in other words, there are no collisions for full duplex or for half duplex there are no collisions and there is no CrS),MAC transmits the preamble and SFD. After that, unless Short Packet transmission is enabled, MAC transmits 64 Bytes of data regardless of the packet length. If the packet length is less than 64 Bytes, the MAC Transmission Block pads the LLC Data field with zeroes. If CRC generation is enabled, the MAC Transmission Block appends CRC to the end of the packet. If a collision occurs during the first 64 Bytes (7-Byte preamble + SFD + 56-Byte frame), the MAC Transmission Block aborts transmission and transmits a jam pattern (a 32-bit sequence of ones). In this case, it advances the transmission retry counter by 1, then transfers control to the Back Off State Machine. After the Back Off time elapses, the MAC Transmission Block tries to resend the packet if there is no problem with the interpacket gap.

If no collisions occur, the MAC Transmission Block transmits the rest of the packet. If the first 64 Bytes are transmitted without any collision, the MAC Transmission Block provides the DMA engine with permission to overwrite these 64 Bytes. After transmitting the 64 Bytes, the MAC Transmission Block transmits the reset of the packet and appends CRC to the end of it. If an underrun occurs in the FIFO or there is a collision that occurs more than 16 times, the MAC Transmission Block makes no attempt to resend the packet and makes preparations to transmit the next packet registered in the queue.

If a transmission error occurs, the MAC Transmission Block sets the appropriate bit of the Transmission Status Register. Also, depending on the Transmission Control Register settings, an interrupt may occur.

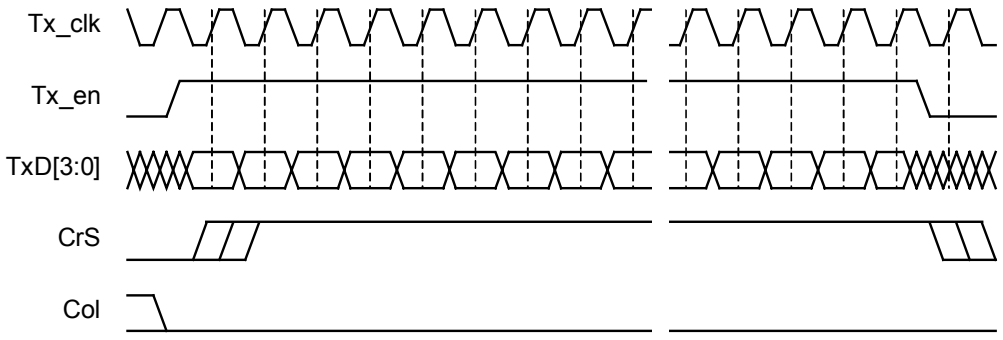


Figure 16.3.10 Transmission with no Collisions

Figure 16.3.10 shows the transmission timing of packets with MII. Tx_end and TxD[3:0] change after the rising edge of Tx_clk, then are stable until sampled at the next rising edge. When in the full duplex transfer mode, CrS and Col become undefined. When in the half duplex transfer mode, CrS must be asserted after Tx_en and must be asserted during packet transmission. If CrS is deasserted during packet transmission, a Carrier Lost error will occur. Col retaining the "L" setting indicates that no collisions have occurred.

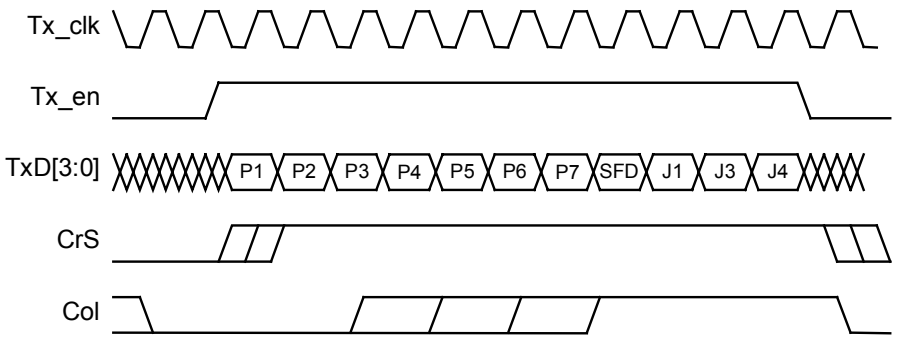


Figure 16.3.11 Transmission when Collision Occurred in the Preamble

Figure 16.3.11 shows the method the Transmission Block uses to handle collisions. It adds 4-Byte jam data before deasserting Tx_en. If a collision occurs in the preamble, the jam data is added on after SFD ends.

16.3.8.5 Frame reception procedure

To receive a frame, the Reception Enable bit (RxEn) of the Reception Control Register must be set, and the Reception Halt Request bit (RxHalt) must be cleared. Furthermore, the Halt Transmission Immediately bit (TxHalt) and the Halt Request bit (HaltReq) of the MAC Control Register must be cleared. Usually, before programming the above registers, the DMA Controller is initialized. This allows the DMA Reception Controller to control transfers from the MAC Reception FIFO. For the method of initializing data structures to enable reception, see the descriptions in subsections 16.3.9 DMA Operation and 16.3.9.5 Receiving frames.

When enabled, the MAC Reception Block always monitors the data stream coming in from MII. When in the Loop-back mode, the data stream is inputted from the MAC Transmission Block via MII.

The MAC Reception Block receives a 0- to 7-Byte preamble and the Start Frame Delimiter (SFD). The MAC Reception Block checks whether the first nibble received is a preamble, then checks whether there is an SFD in the first 8 Bytes of the packet. Except for the preamble, the MAC Reception Block treats a packet that does not have the SFD as its first Byte as a packet fragment and deletes it.

Following after the SFD is the first nibble of the recipient address. The MAC Reception Block generates parity when it receives 1 Byte of data, then stores it with the data in the MAC Reception FIFO. After that, it signals the existence of reception data. The MAC Reception Block collects subsequent nibbles into Byte units and stores them in the appropriate FIFO. The DMA Reception Controller reads them from the MAC Reception FIFO in Byte units, performs a parity check, and then transfers the data to the DMA Reception buffer. When either the MAC Reception FIFO becomes empty or it transmits the last Byte of a packet and ends transmission, the MAC Reception Block signals this fact.

If PHY asserts both the Rx_DV and Rx_er signals while receiving a frame, the MAC Reception Block reports that a CRC error occurred in the current packet.

When the MAC Reception Block receives the recipient address, the ARC tries to recognize the received address. If the ARC refuses a packet, the MAC Reception Block signals this fact and the DMA Reception Block destroys the data packet.

16.3.8.6 Address Recognition Circuit (ARC) operation

To read from or write to ARC memory, first the system software sets the ARC Address Register, and then it reads and writes the ARC Data Register. All Bytes are written to regardless of the Byte Enable status. To write either the upper 2 Bytes or lower 2 Bytes of a double word, the driver software has to handle the data to correctly and simultaneously write 2 adjoining Bytes. The Ethernet Controller does not support Read, Modify, or Write cycles to internal DMA RAM.

Figure 16.3.12 shows how the MAC reads entries from the ARC. ARC entries are in the Big Endian Byte order. In other words, if #0-0 is the first Byte of the first entry, then #0-5 is the sixth and therefore last Byte of the first entry. After entry #20 are the 2 Bytes Rsv-2 and Rsv-3, then the 2 double words MC#1 and MC#2. MC#1 and MC#2 cannot be used by ARC operation, but they are used when generating the MAC control frame. Item 16.3.8.7 Pause operation during full duplex transfer describes this.

Byte 3	Byte 2	Byte 1	Byte 0	
#0-0	#0-1	#0-2	#0-3	00h
#0-4	#0-5	#1-0	#1-1	04h
#1-2	#1-3	#1-4	#1-5	08h
#2-0	#2-1	#2-2	#2-3	0Ch
#2-4	#2-5	#3-0	#3-1	10h
#3-2	#3-3	#3-4	#3-5	14h
#4-0	#4-1	#4-2	#4-3	18h
#4-4	#4-5	#5-0	#5-1	1Ch
#5-2	#5-3	#5-4	#5-5	20h
				...
#18-0	#18-1	#18-2	#18-3	6Ch
#18-4	#18-5	#19-0	#19-1	70h
#19-2	#19-3	#19-4	#19-5	74h
#20-0	#20-1	#20-2	#20-3	78h
#20-4	#20-5	Rsv-2	Rsv-3	7Ch
MC#1-0	MC#1-1	MC#1-2	MC#1-3	80h
MC#2-0	MC#2-1	MC#2-2	MC#2-3	84h

Figure 16.3.12 ARM Memory Map

16.3.8.7 Pause operation during full duplex transfer

16.3.8.7.1 Local Pause operation

To enable Pause operation during full duplex transfer, you have to set a special multicast address for the MAC control frame in ARC memory and set the corresponding bit in the ARC Enable Register. You can store the special multicast address for the MAC control frame anywhere in ARC memory, but there are cases where you will have to be careful of the storage location to optimize the usage efficiency of ARC memory (see 16.3.8.7.2 Remote Pause operation). The MAC reception circuit recognizes full duplex Pause operation when the following conditions are met.

- A particular value (0x8808) is set in the Frame Length/Frame Type field of the MAC control frame.
- The ARC recognizes a packet.
- The packet is 64 Bytes long.
- The Operation field specifies Pause operation (0x0001).

After recognizing full duplex transfer Pause operation, the MAC reception circuit loads the operand values into the Pause Count Register then instructs both the MAC and DMA engines to pause when processing of the current packet ends. If no packet is currently being processed, then both engines immediately pause.

The pause circuit manages the Pause Count Register and counts down the pause time. When the pause time count becomes 0, the pause circuit issues an instruction to end pausing and operation of the transmission circuit resumes.

If another Pause operation is recognized during Pause operation, the Pause Count Register is reset by a new operand value. An operand value of 0 will abort the Pause operation currently in progress.

16.3.8.7.2 Remote Pause operation

The program can freely set the MAC control frame of the Ethernet Controller, so even when performing Pause operation or using MAC control in the future, it will still support this function.

For remote Pause operation or transmission of other MAC control frames, follow the procedure below.

- Set the recipient address to #0 of the ARC memory.
- Set the sender address to #1 of the ARC memory.
- To #20 of the ARC memory set the special MAC control type value (0x0808), the Pause operation opcode (0x0001), and the operand value (pause time: 0x0000 to 0xFFFF). Write 0x0000 to the 2 Bytes after ARC #20.
- Write 0x0000_0000 to MC#1 and MC#2. Then, remote Pause transmission is complete at one bus operation.
- Write to the Transmission Control Register and set the SdPause bit.

Usually, the recipient address is the special multicast address of the MAC control frame and the sender address is the local station address. You can enable these ARC entries to be used for address filtering. Since ARC entry #20 does not include a valid network address, you must not enable it when using it as part of a flow control transmission.

When transmission ends, the transmission status is written in the Transmission Control Frame Status Register. If the End of Transmission Control bit (bit 10) of the Interrupt Enable Control Register is set, the DMA engine will issue interrupts.

16.3.8.8 Error display

The Error flags and Abnormal Operation flags are divided into those that are set during transmission and those that are set during reception. These flags are each set in the Transmission Status Register (Tx_Stat) and the Reception Status Register (Rx_Stat), respectively. Also, the Missing Packet Count Register (Miss_Cnt) contains the number of deleted packets and is used to manage the system network. See 16.4 Registers for the format of these flags and the counter.

16.3.8.8.1 Transmission error display

Transmission operation ends when an entire packet (preamble, SFD, data, CRC) is transferred to a physical medium without any collisions occurring. If an internal error or network error occurs, the MAC Transmission Block reports the content of that error.

If one of the states described below occurs, transmission is aborted and the Status bit is set. After the Status bit is set, an interrupt occurs if the corresponding Interrupt Enable bit in the Transmission Control Register is set.

Table 16.3.4 Transmission Error Display (1/2)

Error Display	Description
MAC Transmission Parity Error	The data that the DMA Transmission Controller transmits to the MAC Transmission FIFO via DII is protected by the Parity bit. If a Parity error occurs, the DMA Transmission Controller halts transmission if the TxParErr bit of the Transmission Status Register is set and interrupts are enabled.
MAC Transmission FIFO Underrun	The MAC Transmission FIFO has a capacity of 80 Bytes so even if a collision occurs, it holds 64 Bytes for retransmission and can support DMA latency up to 1.28 μs (128-bit late time=16-Byte time). The DMA Transmission Controller has sufficient bandwidth, so if an underrun occurs in the MAC Transmission FIFO, this usually indicates that there is a problem with the latency of the PCI Bus. When such an underrun occurs, the Underrun bit of the Transmission Status Register is set.
Carrier Sense Lost	Carrier Sense (CrS) is monitored from the beginning of the Start Frame Delimiter to the last transmission Byte. This error indicates that transmission was not aborted even though CrS either does not exist or was lost due to a network fault or other cause. When in the Loop-back mode, Tx_en drives CrS. With full duplex transfers, Carrier Sense Lost is not asserted since CrS is not handed off to the Transmission Block. When Carrier Sense is lost, the LostCrS bit of the Transmission Status Register is set.
Excessive Collision	If the MAC encounters a collision during transmission, it backs off, updates the collision counter, and then tries to retransmit the data after a specific time interval elapses. When the counter becomes “16”, transmission is aborted if transmission is attempted 16 times and a collision occurred every time. A network fault could cause excessive collisions. When excessive collisions occur, the ExColl bit of the Transmission Status Register is set.

Table 16.3.4 Transmission Error Display (2/2)

Error Display	Description
Late Collision (Collision Outside Window)	When the network is operating properly, the MAC detects a collision in the first 64 Bytes of the transmitted data. When a collision occurs after this time elapses (when one occurs outside the window), this means that there is a network fault. In this case, the LateColl bit of the Transmission Status Register is set and packet transmission is aborted. In other words, transmission is not retried after a late collision occurs.
SQE	When in the 10 Mbps mode, “heartbeat” is checked at the end of the transmitted packet. A heartbeat is a short collision signal that occurs within the first 40-bit late time period after the transmission ends. When a heartbeat is not detected, the SQErr bit of the Transmission Status Register is set.
Defer	This state indicates that it is necessary to defer transmission since the transmission route was already in use when packet transmission was attempted. This is not an error, but unless an error occurs, this state is used to indicate the state of the transmission route. When the Defer state occurs, the TxDefer bit of the Transmission Status Register is set.
Excessive Deferral	There are cases where it is necessary to defer transmission since the transmission route was already in use when the MAC attempted to transmit a packet the first time. If the deferral time is longer than MAX_DEFERRAL (2.4288 ms when in the 10 Mbps mode, 0.24288 ms when in the 100 Mbps mode) and the NoExDef bit of the Transmission Control Register is disabled, then the ExDefer bit of the Transmission Status Register is set. When the Excessive Deferral state occurs, transmission is not aborted, but a network fault may have occurred. It is possible to transmit an Excessive Deferral interrupt before packet transmission is complete and display it in the IntExDefer bit of the Interrupt Factor Register.
Transmission PAUSE	When attempting to transmit a packet, there are cases where the MAC has to defer transmission since the reception of a MAC control frame including Pause operation caused the Transmission Block to stop transmitting. This is not an error, but this state is reported to indicate the network status. Marking the aborted packet sets the Paused bit of the Transmission Status Register for the final packet before Pause becomes effective to complement the software process.

16.3.8.8.2 Reception error display

After detecting the Start Frame Delimiter (SFD), the MAC writes the data received from the physical medium to the MAC Reception FIFO. The MAC Reception Block checks whether an overflow occurred in the MAC Reception FIFO during reception. Also, when reception ends, it checks for external errors (Alignment, CRC, Maximum Frame Length Exceeded).

Table 16.3.5 Reception Error Display

Error Display	Description
MAC Reception Parity Error	Once data enters the MAC Reception FIFO, it is protected by the Parity bit. If the RxParErr bit of the Reception Status Register is set and interrupts are enabled when a Parity error occurs, the MAC Reception Block aborts reception.
Alignment Error	After reception ends, the MAC Reception Block checks whether reception packets are properly framed at the 8-bit boundaries. If they are not properly framed and CRC is invalid, the data is corrupted as it passes along the network and the MAC Reception Block signals an Alignment error. The MAC Reception Block also signals a CRC error at this time. The AlignErr bit and CRCErr bit of the Reception Status Register are set.
CRC Error	After reception ends, if the MAC Reception Block checks CRC and there was an error, it signals that error. The CRC error, Frame Alignment error, and the Maximum Frame Length Exceeded error are network errors that the Reception Block detects. These errors could be detected in the following combinations. • CRC error only • Alignment error and CRC error only • Maximum Frame Length Exceeded error and CRC error only • Alignment error, Maximum Frame Length Exceeded error, and CRC error
Overflow Error	During reception, the data is first stored in the MAC Reception FIFO, and then is transferred to the DMA Reception Controller. If the MAC Reception FIFO becomes full due to a cause such as excessive system latency, then the MAC Reception Block sets the Overflow bit of the Reception Status Register.
Maximum Frame Length Exceeded Error	The MAC Reception Block checks the packet length when reception ends. If it receives a frame that is longer than the maximum frame length of 1518 Bytes when the Long Frame mode is not enabled, the MAC Reception Block signals this error.
MII Error	When PHY detects a media error such as a coding violation, it signals MAC by asserting Rx_er. When MAC acknowledges the assertion of Rx_er, it deletes the received packet. A CRC error is forcibly issued and reception of the packet is terminated. There are also cases where an Alignment error or Minimum Frame Length error are detected.

16.3.8.9 Accessing station management data

Following is the basic sequence of events when accessing station management data.

- The system software checks the Busy bit and confirms that MD is not Busy.
- For a write operation, you have to write data to the Data Register before setting up the Control Register.
- The software writes the MDC address, the Read or Write flag, and sets the Busy bit.
- The Ethernet Controller ends operation then clears the Busy bit.
- In the case of a read operation, the system software can read the Data Register after it detects that the Busy bit was cleared.

16.3.9 DMA operation

This subsection describes the programming of the PCI-mounted Ethernet Controller. The programming details differ slightly depending on whether you select batch processing as the control mode or select Continuous Polling. Unless otherwise stated, the descriptions in this subsection assume the use of Continuous Polling.

Following is the structure of this subsection.

- Initial setup of PCI
- Initial setup of DMA and MAC
- Initializing the queues
- Transmitting frames
- Receiving frames
- Handling interrupts

Note: If you use the SRAM in the DMA Controller without initializing it, Packet errors may be mistakenly detected during packet reception. Implement the following initialization procedure to prevent this phenomenon from occurring.

1) Prepare for initial setup

   Set (write "1" to) the TestMode bit (bit 13) of the DMA_Ctl (0x00) Register.

2) Initialize on-chip memory

   Implement the following on all addresses (0x000-0x3FF) of the on-chip SRAM.

   (i) Set the on-chip SRAM address in ARC_Adr (0x60). (ARC_Loc field: 0x000-0x3FF)

   (ii) Write initialization data 0x0000_0000 to ARC_Data (0x064). (32-bit write)

   You can implement the following to confirm the written data.
   Read ARC_Data(0x64). (32-bit read)

3) Cancel the Test mode

   Reset (write "0" to) 1 in the TestMode bit (bit 13) of the DMA_Ctl (0x00) Register.

16.3.9.1 Initial setup of PCI

When initially setting up the system, you can use the IDSel signal to write to the PCI Configuration Register. This makes it possible for the system to map the Ethernet Controller to a memory boot address space then transfer data to the boot address space to output the IDSel signal.

Registers such as the following require initial setup.

- PCI I/O Base Address Register or PCI Memory Base Address Register: To map the register to an I/O address space or memory space
- PCI Command Register: To customize the PCI function
- There are also situations where the following register requires initial setup.
- PCI Interrupt Register: To customize latency or signal interrupt factors to external pins

16.3.9.2 Initial setup of DMA and MAC

After initially setting up the PCI, the DMA Control Register and MAC Control Register are usually mapped to an I/O address space or memory address space and are available for reading and writing.

Registers such as the following require initial setup.

- DMA Transmission Frame Pointer: To start transmission
- DMA Buffer List Frame Pointer: To provide buffer for receiving data
- DMA Free Descriptor Area Base Register or DMA Free Descriptor Area Size Register: To initialize the reception signaling area
- DMA Transmission Polling Control Register: To customize polling of transmission packets
- DMA Transmission Threshold Register: To customize transmission latency handling
- MAC Transmission Control Register: To change the default transmission settings
- MAC Reception Control Register: To change the default reception settings
- MAC ARC Control Register: To customize the recognition conditions of station addresses and multicast addresses
- MAC ARC Address Register and MAC ARC Data Register: To set the filtering of station addresses and other addresses
- MAC ARC Enable Register: To enable each ARC entry after setting up the ARC

There are also situations where the following registers require initial setup.

- MAC Control Register: To customize the MAC configuration
- DMA Control Register: To customize the Burst size

16.3.9.3 Initializing the queues

The system has to set up the transmission queue, buffer list, and reception descriptor area before starting up the Ethernet Controller.

16.3.9.3.1 Initializing the transmission queue

The Transmission Block has two operation modes: Batch Processing and Continuous Polling. When in the Batch Processing mode, the system software prepares a link list of the frame descriptors to be transmitted. The final descriptor must have "1" set in the EOL (end of list) field. When the last frame descriptor is transmitted, the Transmission Frame Pointer Register reads EOL and transmission ends. To resume transmission after this, the system resets the Transmission Frame Pointer Register.

When in the Continuous Polling mode, the system software also prepares a link list of the frame descriptors to be transmitted. However, the last frame descriptor is a dummy frame descriptor. The beginning of a link list is only a dummy frame descriptor, so it is okay for it to be empty. The system owns the dummy descriptor to prevent the Ethernet Controller from accessing it. Transmitting a new packet overwrites the dummy frame descriptor. This is explained in 16.4.4 Flow Control Register group.

16.3.9.3.2 Initializing the buffer list

The buffer list queue is either a single frame descriptor or multiple frame descriptors combined into a link list containing a list of the empty buffer descriptors that is initialized. It is okay for the buffer list to have one of the following configurations.

1. One frame descriptor containing many free buffer descriptors
2. A link list of frame descriptors
3. Wrap-around queue in which the last frame descriptor points to the first frame descriptor

In configurations 1 and 2 above, the EOL bit of the FDNext field is set while in configuration 3, the FDNext field of the last frame descriptor points to the first frame descriptor. You can use the Reception Fragment Size Register to globally pack a buffer. Also, you can use the Control field (FDCtl) of a frame descriptor to select whether to pack in buffer area units. Setting up the Buffer ID field when packing a buffer is useful in managing memory. For details on packing buffers, see 16.4.3.6 Reception Fragment Size Register, 16.4.3.8 Free Descriptor Area (FDA) Registers, and 16.3.7.2.1 BDCtl field (buffer descriptor control).

16.3.9.3.3 Initializing a reception descriptor area

You can initialize a reception descriptor by writing in the Free Descriptor Area Base Register or the Free Descriptor Area Size Register. The Ethernet Controller starts writing to the reception queue in the reception descriptor area according to these registers.

16.3.9.4 Transmitting frames

Paragraph 16.3.9.3.1 Initializing the transmission queue described transmission in a batch process. For each frame transmission batch, the system initializes the transmission queue and sets a Transmission Frame Pointer Register at the head of the queue.

In the case of continuous polling transmission, the frame descriptor list ends at the dummy frame descriptor owned by the system. The Ethernet Controller enters the Polling mode when it reaches a dummy record. In this mode, the Ethernet Controller periodically checks the frame descriptor Control (FDCtl) field and waits for the system to set the COwnsFD bit. The Transmission Polling Counter Register controls the polling frequency.

To transmit frames in the Continuous Polling mode, the system writes the frame descriptor of the frame to be transmitted at the end of the transmission queue. The system overwrites the old dummy frame descriptor, creates a new dummy frame descriptor, and sets the FDNext field of the old frame descriptor in a new dummy frame descriptor. After that, it sets the COwnsFD bit of the old frame descriptor and transfers ownership to the Ethernet Controller.

16.3.9.4.1 Signaling transmission completion

The system can fetch transmission completion information in various ways.

- Request an interrupt.
- Poll the FDCtl field of the transmitted frame descriptor and confirm system ownership.
- Poll the Transmission Frame Pointer Register.

You can set interrupts to occur either at the end of each frame or at the end of selected frames. When polling the Transmission Frame Pointer Register, it has an invalid value (a value set in FDNext while EOL=1) in the Batch Processing mode or has a dummy frame descriptor address when in the Continuous Polling mode.

16.3.9.5 Receiving frames

For the MAC to receive a frame, the system software has to perform the following operations.

- As described in 16.3.9.3 Initializing the queues, initialize a free buffer list or free descriptor area.
- Write a dummy frame descriptor in a free descriptor area. Set the COwnsFD bit of the FDCtl field so the Ethernet Controller becomes the owner.
- Initialize Free Descriptor Area Base Register to the address of the dummy frame descriptor in the free descriptor area.

There are two ways for the system software to get notification of a reception frame:

- Request an interrupt for each reception frame.
- Poll the dummy frame descriptor and check whether the COwnsFD bit is set.

You can enable interrupts by setting the Reception Complete Interrupt Enable bit of the Reception Control Register.

The system must perform the following processes after it receives a frame.

- Process the frame descriptor, and then release it for reuse.
- Release the buffer returned from the above protocol layer and add it to the free buffer list.

16.3.9.5.1 Processing reception frame descriptors

Free descriptor areas are used by FIFOs. However, the frame processing and time required for returning the corresponding buffer differ depending on the application. Then, the frame descriptor mapped by the Ethernet Controller copies that content to a different area, the frame descriptor of the free descriptor area is released in the order received, and the copied descriptor contents are handed to the upper layer of the protocol stack.

16.3.9.5.2 Releasing buffers

There are two methods of mapping a buffer:

- Start a new frame in a new buffer (Single Frame mode).
- Place several frames or parts of a frame in one buffer (Packed Buffer mode).

The Control field of the Reception Fragment Size Register or Frame Descriptor controls the method of mapping the buffer. The Single Frame mode has the advantage of allowing easy memory management, but has the disadvantage of decreasing the memory usage efficiency. Conversely, the Packed Buffer mode has the advantage when considering memory usage efficiency, but has the disadvantage since memory management becomes complex.

When in the Packed Buffer mode, multiple frames or parts of frames may be placed in the same buffer area, so you must keep several points in mind when managing the memory. The Ethernet Controller counts the number of buffers created in the same buffer area then provides that value as the RxBDSeqN value of the BDCtl field in the buffer descriptor. Then, the system software counts the returned fragments, confirms that all fragments were returned, then releases them.

The buffer ID value (RxBDID) is copied from the buffer descriptor of the free buffer list queue to the buffer descriptor in the reception frame queue. You can use up to 256 buffer IDs. When you require more buffer IDs than that, there are several techniques you can use. For example, you can specify the buffer ID expansion bit in the FDSystem field. You can also use the upper bit of the buffer pointer.

16.3.9.6 Handling interrupts

Interrupts generally use one common interrupt line. To confirm whether this PCI device is the interrupt source, use the system software to read the Interrupt Factor Register. Depending on the content of the Interrupt Factor Register, you may have to read other registers such as the Transmission Status Register or Reception Status Register.

16.4 Registers

16.4.1 Overview

This subsection describes the Ethernet Controller registers you can access. These registers are grouped as follows below.

- PCI Configuration Register Group
- DMA Control, Status Register Group
- Flow Control Register Group
- MAC Control, Status Register Group

During normal operation, once you finish setting them up, few registers require direct access. Transmission/Reception operation is performed using a cyclic queue with a ring-like structure. Control information and status information is transmitted by the data structures described in 16.3.7 Configuration. You have to initialize the DMA Control Registers before starting any transfer operation. You have to access the MAC Registers when it requires a special configuration such as address filtering by the ARC. When using an interrupt drive type, you have to access several of the DMA Registers or MAC Registers from inside the Interrupt Handler, enable or disable interrupts, check the interrupt factors, or clear the Interrupt Condition bit. Also, the Flow Control Register can use a driver to access them in order to monitor the run status of Pause commands issued from a local terminal or remote terminal.

16.4.1.1 Accessing registers

Except for those that use part of RAM, the reserved bits of a register are initialized to "0" or "1". To maintain compatibility even if the method of using the registers changes in the future, do not change the values of the reserved bits when using the software to write to registers.

Also, do not implement programming that depends on the values of the reserved bits.

16.4.1.2 Overview of PCI Configuration Registers

Table 16.4.1 lists the name, mnemonic, address, size, and access type of each PCI Configuration Register. PCI Configuration Registers are used by standard Master/Slave PCI devices. Included in PCI Configuration Registers are Device ID Registers, Control Registers, registers that display status information, and registers that make various settings. These registers are setting during initialization.

Table 16.4.1 PCI Configuration Registers

Address	Mnemonic	Register Name
00h	Vend_ID	Vendor ID Register
02h	Dev_ID	Device ID Register
04h	PCI_Cmd	PCI Command Register
06h	PCI_Stat	PCI Status Register
08h	PCI_Clas	Class Code Register
0Ch	PCI_Ctl	PCI Control Register
10h	IO_BaseA	I/O Base Address Register
14h	MLo_BaseA	Memory Base Address Register
2Ch	Sub_Vend_ID	Subsystem Vendor ID Register
2Eh	Sub_ID	Subsystem ID Register
34h	PCI_Cap_Ptr	PCI Function Pointer
3Ch	PCI_Int	PCI Interrupt Register
40h	PM_Cap	Power Management Function
44h	PM_CSR	Power Management Control Status

The address of a PCI Configuration Register is valid when the input signal IDSel is asserted. The addresses of DMA Registers and MAC Registers are valid when IDSel is not asserted and the upper bits of the addresses match either the I/O Base Address Register (IO_BaseA) or the Memory Base Address Register (MLo_BaseA).

16.4.1.3 Overview of DMA Control, Status Registers

Table 16.4.2 shows the name, mnemonic, address, size, and access type of each DMA Control, Status Register. DMA Control, Status Registers control the transmission queue, reception queue, free buffer list, and free descriptor area. Registers for controlling the fragment size and polling rate are also available.

Table 16.4.2 DMA Control, Status Registers

Address	Mnemonic	Register Name
00h	DMA_Ctl	DMA Control Register
04h	TxFrmPtr	Transmission Frame Pointer
08h	TxThrsh	Transmission Threshold Register
0Ch	TxPollCtl	Transmission Polling Control Register
10h	BLFrmPtr	Buffer List Frame Pointer
14h	RxFragSize	Reception Fragment Size Register
18h	Int_En	Interrupt Enable Register
1Ch	FDA_Bas	Free Descriptor Area Base Register
20h	FDA_Lim	Free Descriptor Area Size Register
24h	Int_Src	Interrupt Function Register

16.4.1.4 Overview of MAC Control, Status Registers

MAC layer registers are broken down into the Flow Control Register and MAC Control, Status Register groups.

Table 16.4.3 shows the name, mnemonic, address, size, and access type of each Flow Control Register.

Table 16.4.3 Flow Control Registers

Address	Mnemonic	Register Name
30h	PauseCnt	Pause Count Register
34h	RemPauCnt	Remote Pause Count Register
38h	TxConFrmStat	Transmission Control Frame Status Register

Table 16.4.4 shows the name, mnemonic, address, size, and access type of each MAC Control, Status Register. Similar to the ARC Control Register and Error Count Register, some registers are accessed by the system software driver when the MAC is activated. A standard DMA engine controls the MAC Transmission Control Register, MAC Transmission Status Register, MAC Reception Control Register, and MAC Reception Status Register after the system software driver sets them up.

Table 16.4.4 MAC Control, Status Register

Address	Mnemonic	Register Name
40h	MAC_Ctl	MAC Control Register
44h	ARC_Ctl	ARC Control Register
48h	Tx_Ctl	Transmission Control Register
4Ch	Tx_Stat	Transmission Status Register
50h	Rx_Ctl	Reception Control Register
54h	Rx_Stat	Reception Status Register
58h	MD_Data	Station Management Data Register
5Ch	MA_CA	Station Management Control, Address Register
60h	ARC_Adr	ARC Address Register
64h	ARC_Data	ARC Data Register
68h	ARC_Ena	ARC Enable Register
7Ch	Miss_Cnt	Missed Error Count Register

MAC layer Control Registers include registers such as the Master MAC Control Register, Transmission/ Reception Control Register, and the ARC Control Register.

16.4.1.5 Register address maps

Figure 16.4.1 through Figure 16.4.4 show the memory addresses of the PCI Configuration Registers, DMA Control, Status Registers, Flow Control Registers, and MAC Control, Status Registers. System memory addresses are listed in the Little Endian mode.

	Byte 3	Byte 2	Byte 1	Byte 0	
	Dev_ID		Vend_ID		00
	PCI_Stat		PCI_Cmd		04
PCI_Clas	Base_Cl	Sub_Cl	Prog_IF	Rev_ID	08
PCI_Ctl	BIST	Hdr_Typ	Lat_Timr	Cache_Sz	0C
	IO_BaseA				10
	MLo_BaseA				14
	Reserved (MHi_BaseA)				18
	Reserved				...
	Sub_ID		Sub_Vend_ID		2C
	Reserved				30
	Reserved			Cap_Ptr	34
	Reserved				38
PCI_Int	Max_Lat	Min_Gnt	Int_Pin	Int_Line	3C
	PM_Cap				40
	PM_CSR				44

Figure 16.4.1 Address Map of PCI Configuration Registers

Byte 3	Byte 2	Byte 1	Byte 0	
Reserved	DMA_Ctl			00
TxFrmPtr				04
Reserved		TxThrsh		08
Reserved		TxPollCtl		0C
BLFrmPtr				10
Reserved		RxFragSize		14
Reserved		Int_En		18
FDA_Bas				1C
Reserved		FDA_Lim		20
Reserved	Int_Src			24

Figure 16.4.2 Address Map of DMA Control, Status Registers

Byte 3	Byte 2	Byte 1	Byte 0	
Reserved		PauseCnt		30h
Reserved		RemPauCnt		34h
Reserved	TxConFrmStat			38h

Figure 16.4.3 Address Map of Flow Control Registers

Reserved	MAC_Ctl		40
Reserved	Reserved	ARC_Ctl	44
Reserved	Tx_Ctl		48
Reserved	Tx_Stat		4C
Reserved	Rx_Ctl		50
Rx_Stat			54
Reserved	MD_Data		58
Reserved	MD_CA		5C
Reserved	ARC_Adr		60
ARC_Data[3:0]			64
Reserved	ARC_Ena[2:0]		68
Reserved	Reserved		6C
Reserved	Reserved		70
Reserved			…
Reserved	Miss_Cnt		7C

Figure 16.4.4 Address Map of MAC Control, Status Registers

16.4.2 PCI Configuration Register group

Each PCI device has a configuration address space totaling 256 Bytes. The PCI specification prescribes the first 72 Bytes of this space in advance. An Ethernet Controller that has the wakeup service only uses the first 74-Byte address space.

16.4.2.1 Vendor ID Register (Vend_ID) 0x00

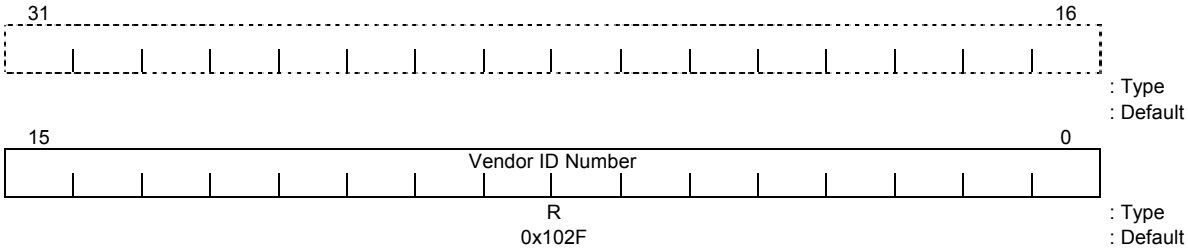


Figure 16.4.5 Vendor ID Register

The Vendor ID Register stores the ID code PCI SIG, which denotes Toshiba as the component manufacturer. The ID code 0xFFFF is reserved and is invalid. Toshiba's PCI SIG number is 0x102F.

16.4.2.2 Device ID Register (Dev_ID) 0x02

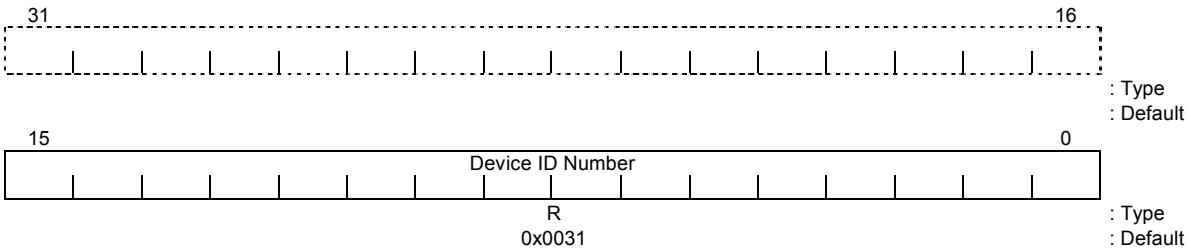
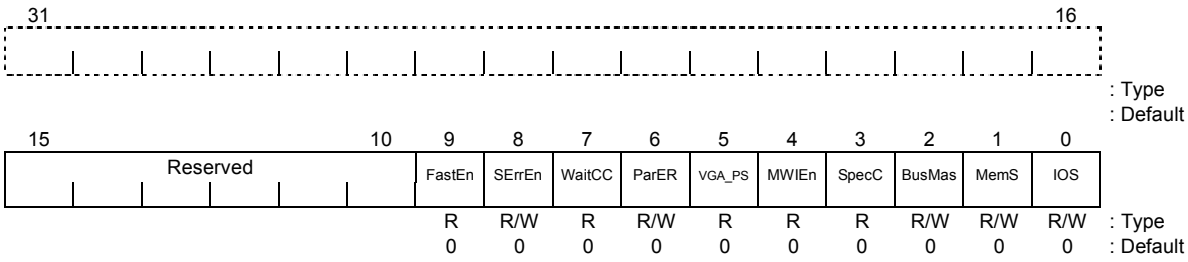


Figure 16.4.6 Device ID Register

The Device ID Register stores a 16-bit value for identifying a particular device. The device ID number Toshiba allocated is 0x0031. The Class Code Register stores the device's revision ID.

16.4.2.3 PCI Command Register (PCI_Cmd) 0x04



Bit(s)	Mnemonic	Field Name	Description
15:10		Reserved	
9	FastEn	Fast Back-to-back Enable	FastEn (fixed to "0", R) The Ethernet Controller cannot perform two consecutive bus transactions.
8	SErrEn	System Error Enable	SerrEn (Default: 0, R/W) Enables the system error (SERR#) driver.
7	WaitCC	Wait Cycle Control	WaitCC (fixed to "0", R) The Ethernet Controller does not generate address/data stepping.
6	ParER	Parity Error Response	ParER (Default: 0, R/W) The device responds to a parity error.
5	VGA_PS	VGA Palette Snoop	VGA_PS (fixed to "0", R) Does not perform special VGA Palette Snoop.
4	MWIEn	Memory Write and Invalidate Enable	MWIEn (Default: 0, R) The Ethernet Controller does not issue the Memory Write and Invalidate command.
3	SpecC	Special Cycle	SpecC (fixed to "0", R) The Ethernet Controller ignores special cycles.
2	BusMas	Bus Master	BusMas (Default: 0, R/W) The device can operate as a Bus Master.
1	MemS	Memory Space	MemS (Default: 0, R/W) The device responds to memory access.
0	IOS	I/O Space	IOS (Default: 0, R/W) The device responds accesses to the I/O space.

Figure 16.4.7 PCI Command Register

Hardware reset initializes the PCI Command Register to 0x0000. Software reset does not initialize the register contents.

The PCI Command Register defines the method of generating and responding to PCI cycles. For details, see Subsection 6.2.2 Device Control in the PCI specification.

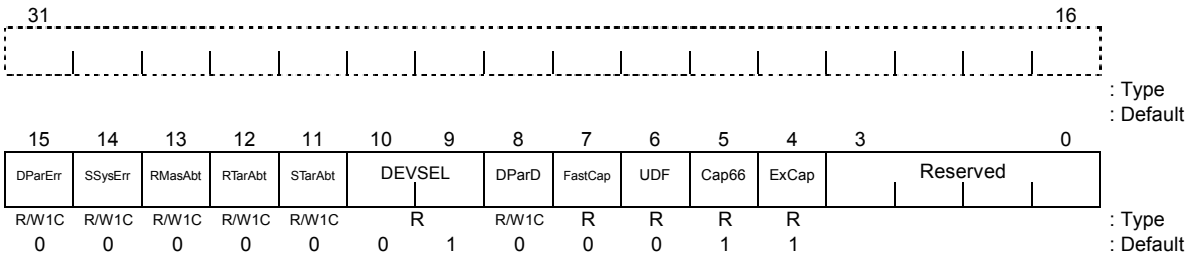
For the Ethernet Controller to function properly, you have to set the BusMas bit and MemS bit or IOS bit.

Set the SErrEn bit and the ParER bit to match the system.

The FastEn, WaitCC, VGA_PS, MWIEn and SpecC bits are always "0". The Ethernet Controller ignores any writes to these bits.

The Ethernet Controller has only one address comparator. The address comparator shares memory access and I/O access as described below. If MemS=0 and IOS=1, the Ethernet Controller responds to any accesses made to the I/O space. If MemS=0 and IOS=0, the Ethernet Controller does not respond to any accesses made to the I/O space or to the memory space. If MemS=1, then the Ethernet Controller only responds to accesses made to the memory space regardless of the IOS value. Also, you have to initialize the corresponding base address registers in addition to MemS and IOS.

16.4.2.4 PCI Status Register (PCI_Stat) 0x06


31
16
: Type
: Default
15 14 13 12 11 10 9 8 7 6 5 4 3 0
DParErr SSysErr RMasAbt RTarAbt STarAbt DEVSEL DParD FastCap UDF Cap66 ExCap Reserved
R/W1C R/W1C R/W1C R/W1C R/W1C R R/W1C R R R R R
0 0 0 0 0 0 1 0 0 0 1 1
: Type
: Default

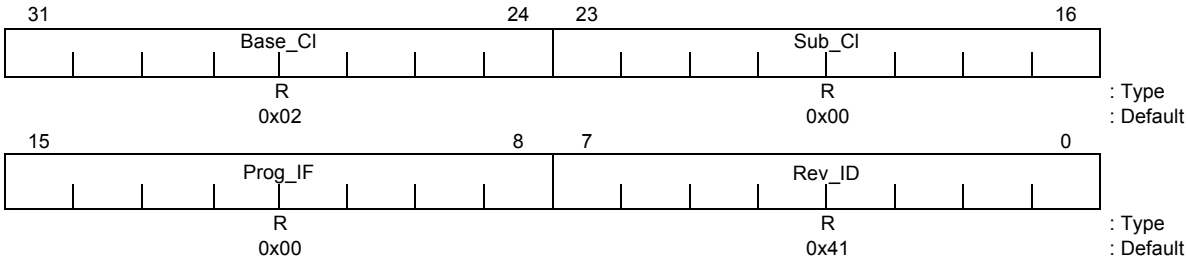
Bit(s)	Mnemonic	Field Name	Description
15	DParErr	Detected Parity Error	DParErr (Default: 0, R/W) Indicates that a parity error was detected. This bit is set if a parity error is detected even when the register's Parity Error Response bit is not set. This also applies to parity errors occurring during an address cycle.
14	SSysErr	Signal System Error	SSysErr (Default: 0, R/W) This bit is set when the device asserts SERR#.
13	RMasAbt	Receive Master Abort	RMasAbt (Default: 0, R/W) Indicates that the current device is the Bus Master and that Master Abort ended the Bus Master transaction (excluding special cycles).
12	RTarAbt	Receive Target Abort	RTarAbt (Default: 0, R/W) Indicates that the current device is the Bus Master and that Target Abort ended the Bus Master transaction.
11	STarAbt	Signal Target Abort	STarAbt (Default: 0, R/W) Indicates the current device is the target and that Target Abort ended the Bus Master transaction.
10:9	DEVSEL	Device Select Timing	DEVSEL (Fixed to "01", R) The latest output timing of DEVSEL# when the Ethernet Controller is accessed as the target device is "medium speed". This field is encoded as follows. 00: Fast 01: Medium Speed 10: Slow
8	DParD	Detect Data Parity Error	DParD (Default: 0, R/W) This bit is set when all of the three following conditions are met. 1: PERR# was asserted as the Bus Master, but the target asserted it. 2. The agent that set PERR# was the Bus Master. 3. The Parity Error Response bit of the PCI Control Register is set to "1".
7	FastCap	Fast Back-to-back Capable	FastCap (fixed to "0", R) The Ethernet Controller cannot perform consecutive transactions from different agents.
6	UDF	User Defined Function	UDF (fixed to "0", R) Indicates that there are no user-defined functions in the Ethernet Controller.
5	Cap66	66 MHz Operation Capable	Cap66 (fixed to "1", R) Indicates that the Ethernet Controller can operate at 33-66 MHz.
4	ExCap	Extended Function	ExCap (fixed to "1", R) Indicates that the Ethernet Controller has extended functions.
3:0		Reserved	

Figure 16.4.8 PCI Status Register

Software reset initializes the PCI Status Register to 0x0230.

The PCI Status Register retains status information on events relating to the PCI Bus. There are several points about this register that you should keep in mind. First, writing "1" to a bit in this register clears it to "0". Also, the value of each bit does not change when you write "0" to it. This enables the system and PCI device to simultaneously update the status information. For details, see Subsection 6.2.3 Device Status of the PCI specification.

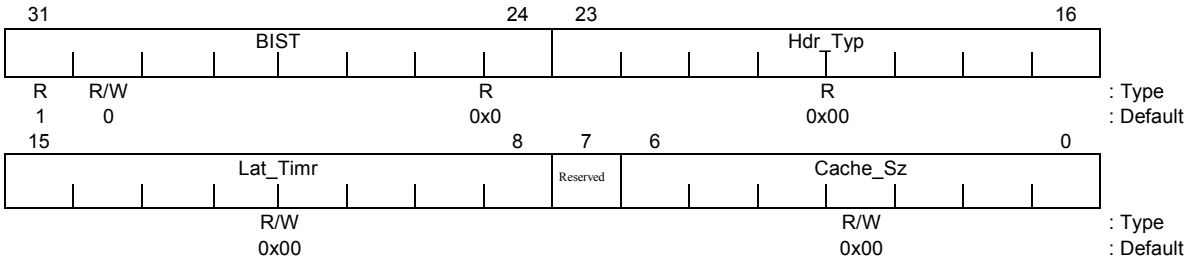
16.4.2.5 Class Code Register (PCI_Clas) 0x08



Bit(s)	Mnemonic	Field Name	Description
31:24	Base_Cl	Base Class ID	Base_Cl (fixed to "0x02", R)
23:16	Sub_Cl	Subclass ID	Sub_Cl (fixed to "0x00", R) Is defined in the Ethernet Controller.
15:8	Prog_IF	Programming Interface	Prog_IF (fixed to "0x00", R) The register-level programming interface is not defined.
7:0	Rev_ID	Revision ID	Rev_ID (fixed to "0x41", R) Is set to "0x41" for this Ethernet Controller.

Figure 16.4.9 Class Code Register

16.4.2.6 PCI Control Register (PCI_Ctl) 0x0C



Bit(s)	Mnemonic	Field Name	Description
31:24	BIST	Embedded Self-Test	BISAT (Default: 0x80, R/W) Controls whether to invoke BIST during startup.
23:16	Hdr_Typ	Header Type	Hdr_Typ (fixed to "0x00", R) This is a single-function device. The range 0x10-0x3F of the configuration space is the standard layout.
15:8	Lat_Timr	Latency Timer	Lat_Tmr (DefaultA: 0x00, R/W) Sets the time the Ethernet Controller operates as the Bus Master as a PCI Bus Clock count.
7		Reserved	
6:0	Cache_Sz	Cache Line Size	Cache_Sz (Default: 0x00, R/W) Sets the system cache line size.

Figure 16.4.10 PCI Control Register

Hardware reset initializes the PCI Control Register to 0x8000_0000. Software reset does not change the register contents.

BIST is used for testing the buffers on the chip. Bit 31 of the BIST field is read only and is fixed to "1". In other words, the Ethernet Controller supports BIST. Bit 30 is used to start invoking BIST. Writing "1" to bit 30 starts a test. This bit is cleared when a test ends. If an error occurs in a test, then either bit 25 or bit 24 is set. Bit 25 is set if a parity error occurs when starting up SRAM inside DMA. Bit 24 indicates that there is a fault in the SRAM in DMA. In other words, it indicates that the read data and the expected value (read data) do not match. Bits 29:26 are reserved.

The software driver invokes BIST during initialization. (Note: Invoking BIST affects operation since RAM data and RAM registers are overwritten.)

When the clock is 33 MHz, testing 1 K × 4-Byte memory requires approximately 123 μs.

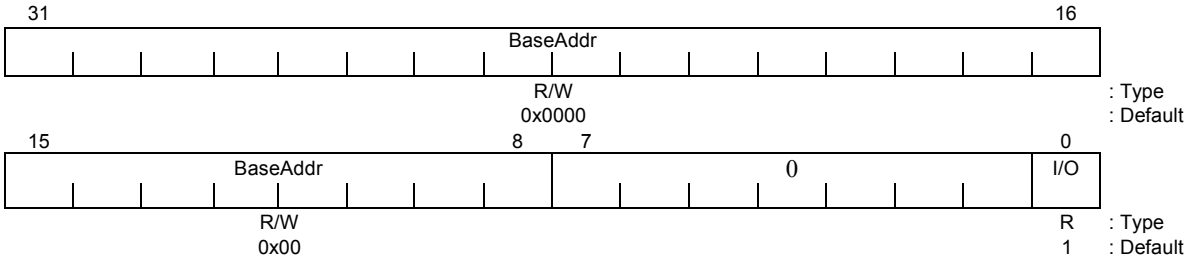
The default value of the latency timer is 0. You can use a program to set the latency timer.

After performing a hardware reset, the cache line size is initialized to 0. You must use the software driver to set an appropriate default value. In most cases, the recommended value is 8 double words (32 Bytes). The maximum value of the cache line size is 127 double words. The cache line size is used to select the Memory Read Multiple command or Memory Read Line command and perform Burst reads.

16.4.2.7 I/O, Memory Base Address Registers

I/O Base Address Register (IO_BaseA) 0x10

The I/O Base Address Register and Memory Base Address Register are used to map the DMA and MAC Control Registers or Status Registers to the I/O address space or system memory space. Both the I/O address space and memory address space have a maximum size of 32 bits. In addition to setting the Base Address Register, you have to set the corresponding Control bits in the PCI Command Register.

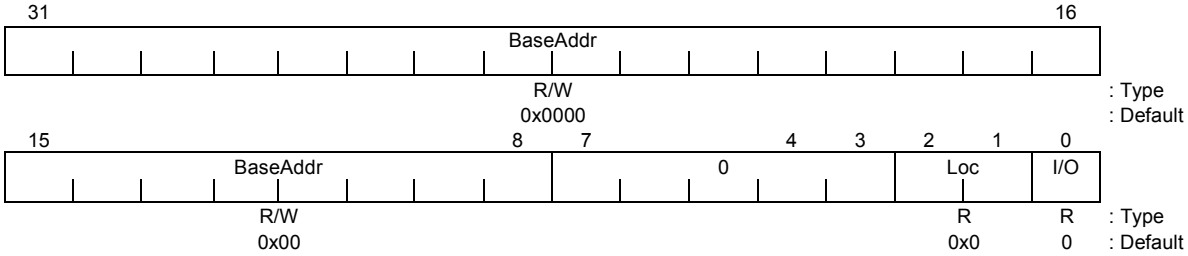


Bit(s)	Mnemonic	Field Name	Description
31:8	BaseAddr	Base Address	BaseAddr (Default: 0x000000, R/W) This field sets the upper 24 bits of the base address
7:1			Fixed to “0”.
0	I/O	I/O Flag	I/O (fixed to “1”, R) Indicates that this base address is relative to the I/O space.

Figure 16.4.11 I/O Base Address Register

Hardware resets initialize the I/O Base Address Register to 0x0000_0001. Software resets do not change the register contents.

Memory Address Register (MLo_BaseA) 0x14



Bit(s)	Mnemonic	Field Name	Description
31:8	BaseAddr	Base Address	BaseAddr (Default: 0x00_0000, R/W) This field sets the upper 24 bits of the base address.
7:3			Fixed to “0”.
2:1	Loc	Location Bit	Loc (Default: 00, R) This field indicates that memory is placed in a 32-bit address space.
0	I/O	I/O Flag	I/O (Default: 1, R) Indicates that this base address is relative to the memory space.

Figure 16.4.12 Memory Base Address Register

Hardware resets initialize the Memory Base Address Register to 0x0000_0000. Software resets do not change the register contents.

16.4.2.8 Subsystem Vendor ID Register, Subsystem ID Register

Subsystem Vendor ID Number Register (Sub_Vend_ID) 0x2C

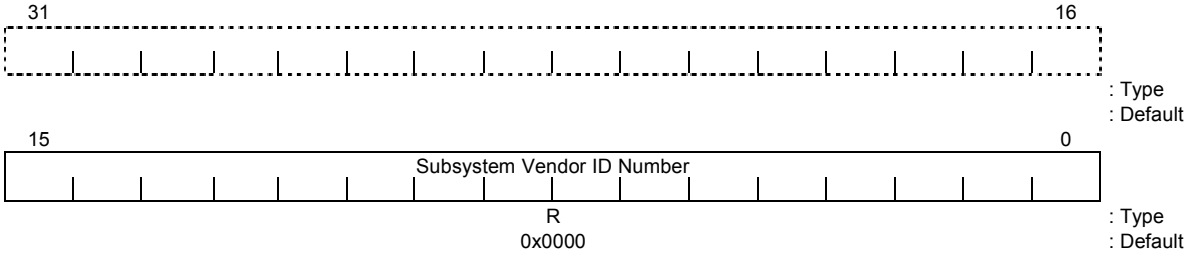


Figure 16.4.13 Subsystem Vendor ID Number Register

The Subsystem Vendor ID Register stores the PCI SIG ID number that denotes the adapter card manufacturer. In the case of the TX4938, this value is 0 since it uses an on-chip PCI device.

Subsystem ID Register (Sub_ID) 0x2E

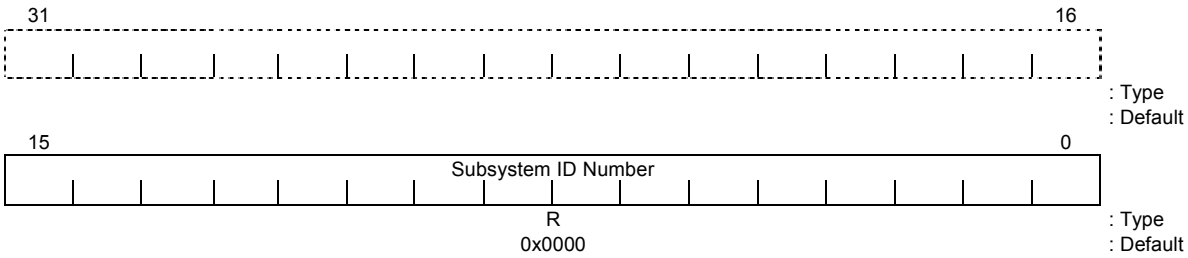
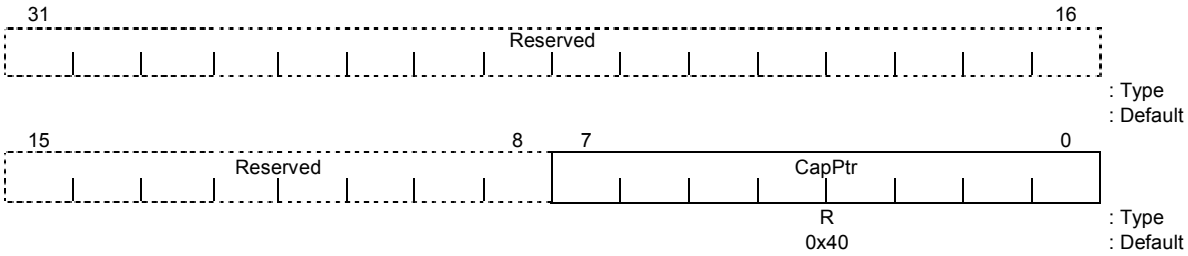


Figure 16.4.14 Subsystem ID Register

The Subsystem ID Register stores a 16-bit value for identifying a particular device. In the case of the TX4938, this value is 0.

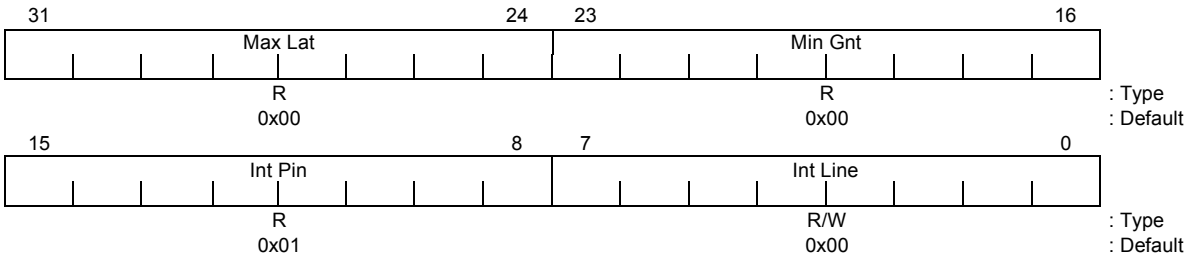
16.4.2.9 PCI Function Pointer Register (PCI_Cap_Ptr) 0x34



Bit(s)	Mnemonic	Field Name	Description
31:8		Reserved	
7:0	CapPtr	Function Pointer	CapPtr (Fixed to “0x40”, R) This field is the offset of the first entry in the function list.

Figure 16.4.15 PCI Function Pointer Register

16.4.2.10 PCI Interrupt Register (PCI_Int) 0x3C



Bit(s)	Mnemonic	Field Name	Description
31:24	Max_Lat	Maximum Latency	Max_Lat (Fixed to “0x00”, R) Sets the latency timer value (unit: 1/4 μs).
23:16	Min_Gnt	Minimum Grant	Min_Gnt (Fixed to “0x00”, R) Assuming the PCI clock is 33 MHz, this field sets the time required for Burst transfer (unit: 1/4 μs).
15:8	Int_Pin	Interrupt Pin	Int_Pin (Fixed to “0x01”, R) Is set to INTA#.
7:0	Int_Line	Function Pointer	Int_Line (Default: 0x00, R/W) Is set by the system. Is the connection information of the interrupt line.

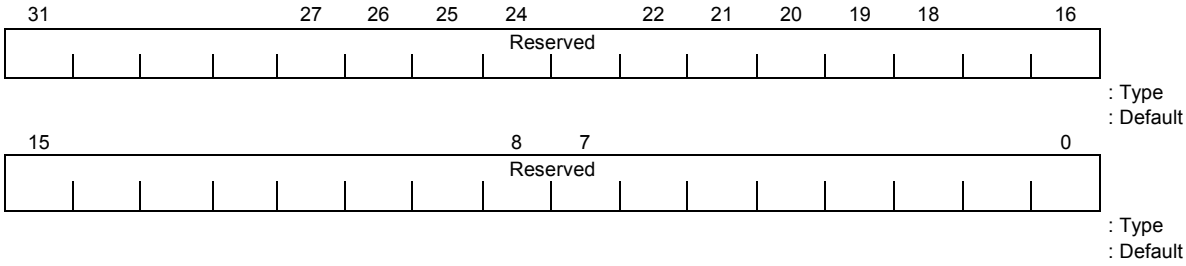
Figure 16.4.16 PCI Interrupt Register

When the value of Max_Lat and Min_Gnt is 0, you do not use the PCI Interrupt Register to determine Maximum Latency and Minimum Grant.

To check whether this PCI device caused an interrupt, use the software to access the Interrupt Source Register. For details, see 16.4.3.9 Interrupt Source Register.

16.4.2.11 PCI Power Management Registers

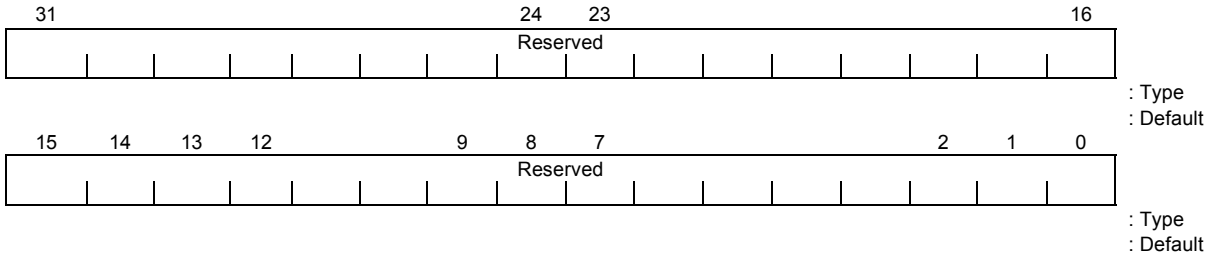
Power Management Function Register (PM_Cap) 0x40



Bit(s)	Mnemonic	Field Name	Description
31:0		Reserved	Reserved The Ethernet Controller does not support the Power Management Function Register.

Figure 16.4.17 Power Management Function Register

Power Management Control, Status Register (PM_CSR) 0x44



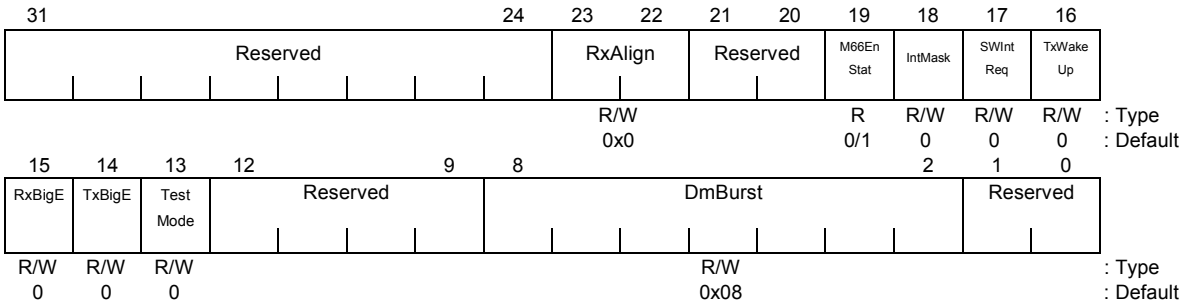
Bit(s)	Mnemonic	Field Name	Description
31:0		Reserved	Reserved The PCI Controller does not support the Power Management Control, Status Register.

Figure 16.4.18 Power Management Control, Status Register

16.4.3 DMA Control, Status Register group

The DMA engine and system software jointly manage three queues: the transmission queue, the reception queue, and the buffer list. The transmission queue is used for frame descriptors that are ready for transmission and are standing by. The reception queue is used for frame descriptors that have been received and are waiting for processing by the system software. The buffer list is a buffer descriptor queue. Buffer descriptors describe system memory areas that can be used to store reception data. The free descriptor area (FDA) is a memory area that the Ethernet Controller can write transmission queue frame descriptors and buffer list descriptors to.

16.4.3.1 DMA Control Register (DMA_Ctl) 0x00



Bit(s)	Mnemonic	Field Name	Description
31:24		Reserved	
23:22	RxAlign	Reception Alignment	D2Supp (Default: 0x0, R/W) This field controls the alignment of reception packets as follows: 01: Skips the first byte of the first buffer. 10: Skips the first or second byte of the first buffer. 11: Skips the first through third byte of the first buffer.
21:20		Reserved	
19	M66EnStat	66 MHz Enable State	M66EnStat (Fixed to 0/1, R) 1: Indicates that the PCI Bus operates at a maximum of 66 MHz. 0: Indicates that the PCI Bus operates at 33 MHz or less.
18	IntMask	Interrupt Mask	IntMask (Default: 0, R/W) Setting this bit to “1” disables the interrupt request signal.
17	SWIntReq	Software Interrupt Request	SWIntReq (Default: 0, R/W) Setting this bit to “1” issues an interrupt request.
16	TxWakeUp	Transmission Wake Up	TxWakeUp (Default 0, R/W) Setting this bit to “1” ends the current polling cycle and starts transmission.
15	RxBigE	Reception Big Endian	RxBigE (Default: 0, R/W) Setting this bit to “1” processes reception data in the Big Endian mode.
14	TxBigE	Transmission Big Endian	TxBigE (Default: 0, R/W) Setting this bit to “1” processes transmission data in the Big Endian mode.
13	TestMode	Test Mode	TestMode (Default: 0, R/W) Enables the test mode function.
12:9		Reserved	
8:2	DmBurst	DMA Burst Size	DmBurst (Default: 0x08, R/W) Indicates the Burst size of data transfer executed in the Master mode.
1:0		Reserved	

Figure 16.4.19 DMA Control Register

Hardware resets initialize the DMA Control Register as follows:

- 0000_1020h: When the PCI Bus speed is 0-33 MHz
- 0008_1020h: When the PCI Bus speed is 33-66 MHz

0x20 of the lower byte denotes the default (32 Bytes = 8 double words) of the DMBurst field.

The DMA Control Register controls data transfer functions in the Master mode such as the Burst side, Big Endian handling, and the test mode. This register also controls various DMA functions during transmission such as wake up and software interrupts.

The DmBurst field controls the data transfer size with which the PCI Bus was used when operating in the Master mode. Bits 8:0 set the data transfer size (DMA Burst size), but the lower 2 bits are fixed to "0" and must be a multiple of 4. After a hardware reset, the default value becomes 32 Bytes, in other words 8 double words. You can change this value using a software driver. You cannot set the DmBurst field to "0". Writing of "0" to this field is ignored. Generally, a multiple of the PCI cache line size is set in the DmBurst field. If the Burst size is 4, 8 or 12 when in the 100 Mbps full duplex mode, you have to take into consideration throughput reduction of the PCI Bus.

The TestMode bit enables test functions such as those that enable reading or writing to all areas of the internal DMA buffer or those that display internal status information in the reserved bits of a register.

The TxBigE bit and RxBigE bit enable the transmitting or receiving of data with Big Endian mode devices. However, it is important to note that only the data (bytes in the area designated by the buffer descriptor) is handled in the Big Endian mode. Control information such as frame descriptors and buffer descriptors are always in a format unique to the PCI Bus. In other words, they are in the Little Endian format.

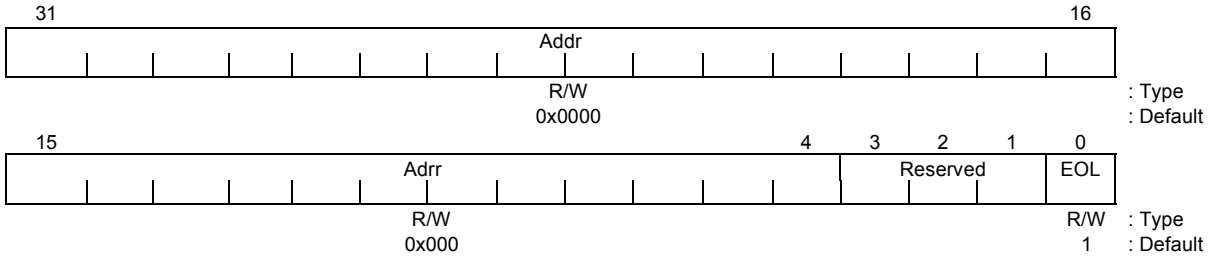
The TxWakeUp bit enables immediate data transmission without waiting for the end of the current polling cycle. Setting the TxWakeUp bit to "1" while the transmitter is polling aborts the current polling cycle. When the current polling cycle ends, the current TxWakeUp bit is cleared. Any writing of "0" to the TxWakeUp bit is ignored.

Software interrupts are made available to support software drivers.

The IntMask bit disables all interrupt sources. Therefore, even when the processing of interrupt sources is in progress, the software driver can put interrupts in the Enable state again.

The M66EnStat bit is used to control the MAC clock divide circuit when driving serial addresses of the MII station manager.

16.4.3.2 Transmission Frame Pointer Register 0x04



Bit(s)	Mnemonic	Field Name	Description
31:4		Address	Addr (Default: 0x000_0000, R/W) Retains the address of the first frame descriptor to be transmitted.
3:1		Reserved	
0	EOL	End of List	EOL (Default: 1, R/W) When this bit is set to “1”, the Address field is ignored. The Ethernet Controller waits for the system to clear it.

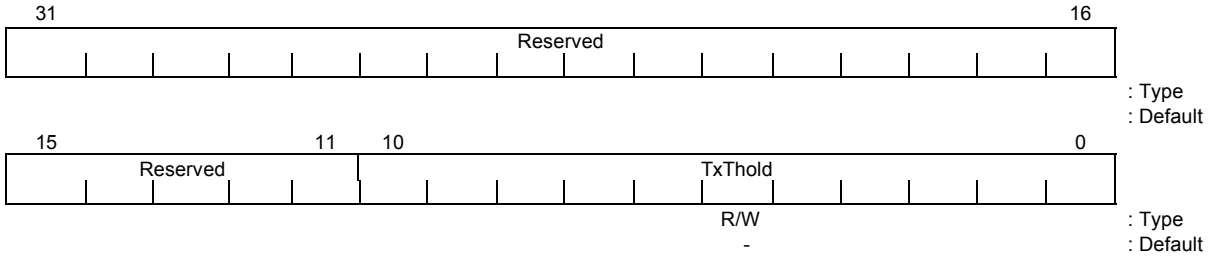
Figure 16.4.20 Transmission Frame Pointer Register

Software resets initialize the Transmission Frame Pointer Register to 0x0000_0001.

Software resets set the EOL bit to “1”. To enable polling of a transmission or the packet to be transmitted, the system has to set this register in a properly initialized frame descriptor. The address has to be aligned to a 16-Byte boundary. Therefore, bits 0-3 must be “0”.

For information on polling control methods, see the description in 16.4.3.4 Transmission Polling Control Register.

16.4.3.3 Transmission Threshold Register (TxThrsh) 0x08



Bit(s)	Mnemonic	Field Name	Description
31:11		Reserved	
10:0	TxThold	Transmission Threshold Register	TxThold (Default: -, R/W) The Transmission Threshold Register controls buffer latency when transmitting packets.

Figure 16.4.21 Transmission Threshold Register

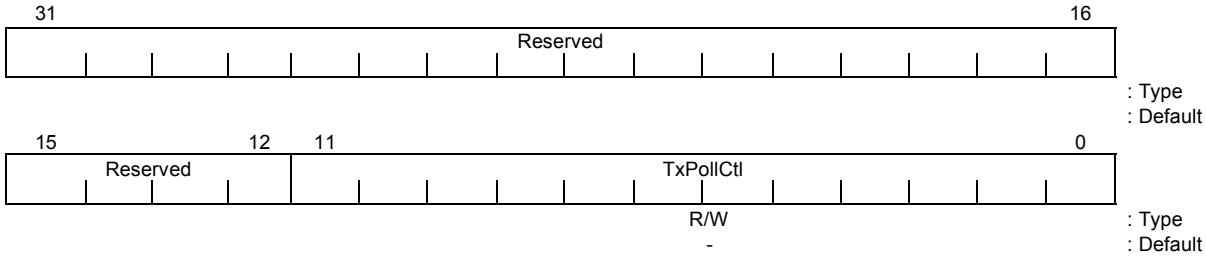
The Transmission Threshold Register uses part of internal RAM, so it is not affected by hardware resets or software resets.

The Transmission Threshold Register controls buffer latency when transmitting packets. If the threshold value is not “0”, data transfer to the MAC starts either when the number of data bytes set by the threshold value (TxThold) has accumulated in the DMA Transmission buffer or when an entire packet is stored in the DMA Transmission buffer. When the threshold value is “0”, data is transferred to the MAC immediately after the data is fetched from the PCI Bus. The software driver initializes the Transmission Threshold Register.

When the threshold value (TxThold) is too small, the DMA Transmission buffer immediately becomes free due to the PCI Bus latency. This situation is displayed in the MAC transmission status, so the system software increases the threshold value.

Be careful to not make the threshold value greater than 1620. Having a threshold value greater than 1620 when in the Long Packet mode causes buffer memory to become full before transmission is enabled, therefore causing the transmission circuit to hang.

16.4.3.4 Transmission Polling Control Register (TxPollCtl) 0x0C



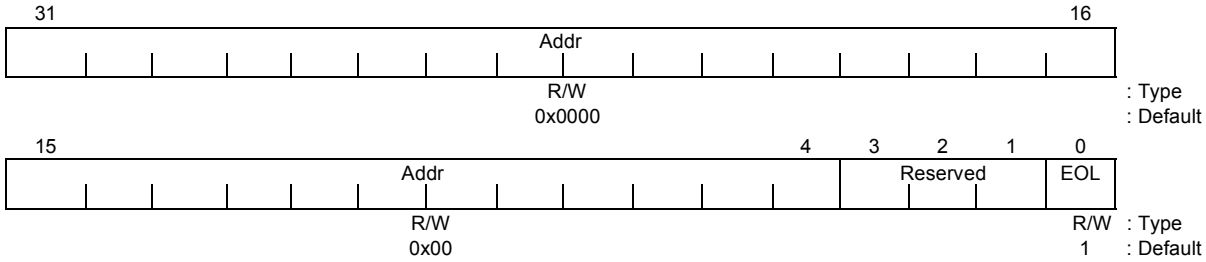
Bit(s)	Mnemonic	Field Name	Description
31:12		Reserved	
11:0	TxPollCtl	Transmission Polling Control Register	TxPollCtl (Default: -, R/W) The Transmission Polling Control Register controls the frequency at which packets to be transmitted are polled.

Figure 16.4.22 Transmission Polling Control Register

The Transmission Polling Control Register uses part of internal RAM, so it is not affected by hardware resets or software resets.

The Transmission Polling Control Register controls the frequency at which packets to be transmitted are polled. The TxPollCtl field operates as a counter. When a value is set to it, it decrements until it reaches 0. When it reaches 0, it polls to see if preparations for the next transmission packet are complete. No polling is performed when 0 is set in the TxPollCtl field. When the clock frequency is 33 MHz, the counter units equate to 61.44 μs. The software driver initializes this register. The software driver also sets the count value.

16.4.3.5 Buffer List Frame Pointer (BLFrmPtr) 0x10



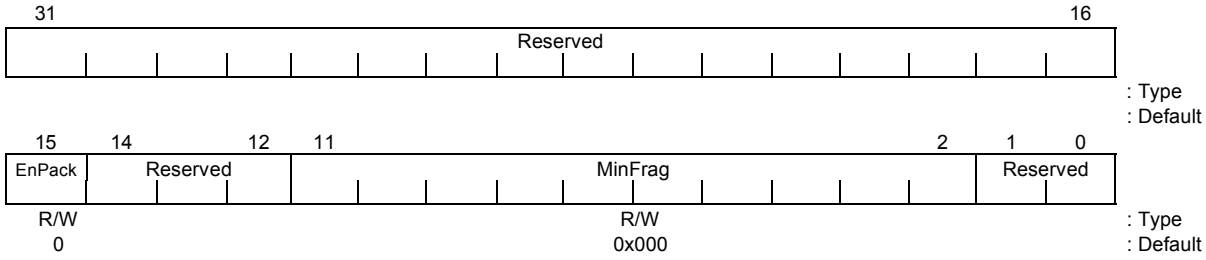
Bit(s)	Mnemonic	Field Name	Description
31:4		Address	Addr (Default: 0x000_0000, R/W)
4:1		Reserved	
0	EOL	End of List	EOL (Default: 1, R/W) When this bit is set to “1”, the Address field is ignored. You have to wait until the system clears this bit.

Figure 16.4.23 Buffer List Frame Pointer

Software resets initialize the Buffer List Frame Pointer to 0x0000_0001.

When it fetches a free buffer descriptor, the Buffer List Frame Pointer holds the address of the first frame descriptor to be read. To enable data reception, the system has to set this register in a properly initialized frame descriptor. The address must be aligned to a 16-Byte boundary. In other words, bits 0-3 must be “0”.

16.4.3.6 Reception Fragment Size Register (RxFragSize) 0x14



Bit(s)	Mnemonic	Field Name	Description
31:16		Reserved	
15	EnPack	Enable Packing	EnPack (Default: 0, R/W) 1: Uses the MinFrag value for buffer packing control. 0: Uses the FDCtl field of the frame descriptor for buffer packing control.
14:12		Reserved	
11:2	MinFrag	Minimum Fragments	MinFrag (Default: 0x000, R/W) The minimum byte count when partially writing to a buffer that contains data.
1:0		Reserved	

Figure 16.4.24 Reception Fragment Size Register

Hardware resets initialize the Reception Fragment Size Register to 0x0000_0000. Software resets do not change the register contents.

The Reception Fragment Size Register specifies the size of minimum data fragments that the Ethernet Controller generates. The minimum fragment size must be a multiple of 4. Therefore, the lower 2 bits are always “0”. You can use the EnPack bit to globally enable packing, or you can enable packing in buffer-area units. For more information on the enabling of packing in buffer-area units, see the description in 16.3.7.1.5 FDCtl field (frame descriptor control).

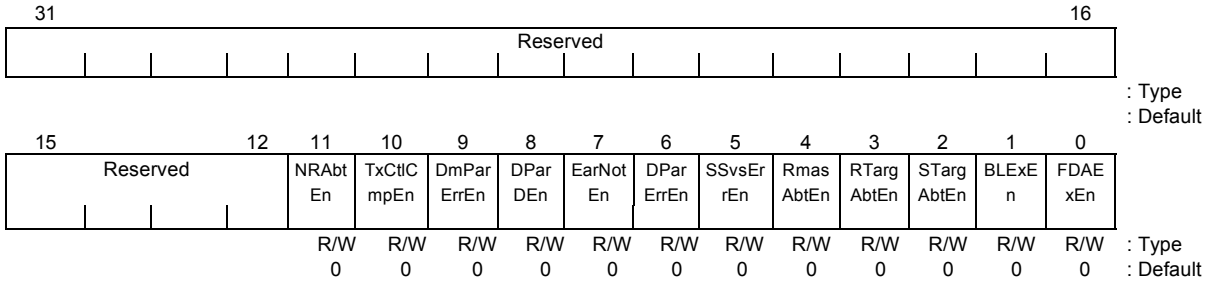
The Ethernet Controller always stores reception data in addresses aligned to 4-Byte boundaries. Therefore, the last 1-3 bytes of a frame may be unused.

When enabling packing, the MinFrag value must be greater than 0 for the Ethernet Controller to function properly. When using packing, use a software driver to set the MinFrag field and EnPack bit.

When not enabling packing, the MinFrag value must remain at “0”.

Packing is performed when more than the Minimum Fragment Size + 4 Bytes in the buffer is free. Packing is not performed when the amount of free space in the buffer is less than the Minimum Fragment Size. In this case, data is stored starting from the next buffer.

16.4.3.7 Interrupt Enable Register (Int_En) 0x18



Bit(s)	Mnemonic	Field Name	Description
31:12		Reserved	
11	NRAbtEn	Non-recoverable Abort Enable	NRAbtEn (Default: 0, R/W) Enables interrupts when a non-recoverable abort occurs internally.
10	TxCtlCmpEn	MAC Control Frame Transmission Complete Enable	TxCtlCmpEn (Default: 0, R/W) Enables interrupts when transmission of the MAC control frame is complete.
9	DmParErrEn	DMA Parity Error Enable	DmParErrEn (Default: 0, R/W) Enables interrupts if a parity error is detected when reading or writing from/to DMA-internal RAM.
8	DParDEn	Data Parity Detection Enable	DParDEn (Default: 0, R/W) Enables interrupts when bit 8 of the PCI Status Register is set.
7	EarNotEn	Early Notification Enable	EarNotEn (Default: 0, R/W) When receiving a reception packet, issues an interrupt not only when reaching the tail of a packet, but also issues an interrupt when writing the first buffer or its descriptor.
6	DParErrEn	Parity Error Detection Enable	DParErrEn (Default: 0, R/W) Enables interrupts if a parity error is detected during PCI Bus transfer while the Ethernet Controller is accessing the Bus Master.
5	SSvsErrEn	System Error Notification Enable	SSvsErrEn (Default: 0, R/W) Enables interrupts if the Ethernet Controller signals a system error.
4	RmasAbtEn	Master Abort Reception Enable	RmasAbtEn (Default: 0, R/W) Enables interrupts if a Master Abort is received while the Ethernet Controller is operating as the Target.
3	RTargAbtEn	Target Abort Reception Enable	RTargAbtEn (Default: 0, R/W) Enables interrupts if a Target Abort is received while the Ethernet Controller is operating as the Target.
2	STargAbtEn	Target Abort Notification Enable	STargAbtEn (Default: 0, R/W) Enables interrupts if a Target Abort is issued while the Ethernet Controller is operating as the Target.
1	BLExEn	Buffer List Exhaustion Notification Enable	BLExEn (Default: 0, R/W) Enables interrupts when the buffer list is completely used up. In other words, enables interrupts when the Ethernet Controller encounters descriptors owned by the system that still remain in the buffer list.
0	FDAExEn	Free Descriptor Area Exhaust Notification Enable	FDAExEn (Default: 0, R/W) Enables interrupts when the free descriptor area is totally used up. In other words, enables interrupts when the Ethernet Controller encounters blocks owned by the system that still remain in the FDA.

Figure 16.4.25 Interrupt Enable Register

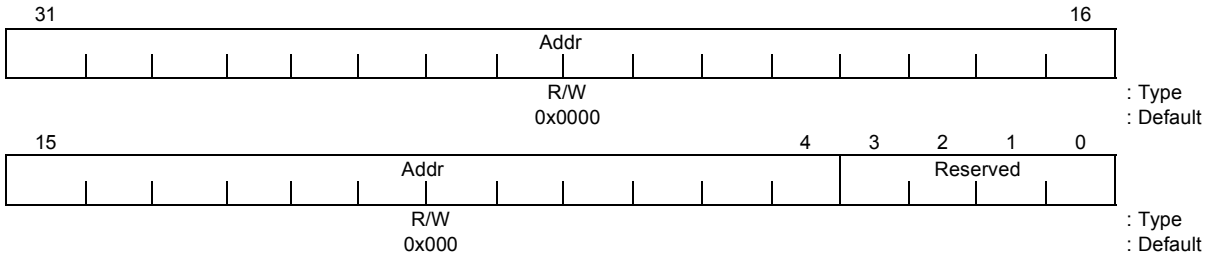
Hardware resets initialize the Interrupt Enable Register to 0x0000_0000. Software resets do not change the register contents.

The Interrupt Enable Register controls whether to issue an interrupt in response to errors detected by the DMA engine or in response to some other conditions.

The Early Notification Enable (EarNotEn) bit is shared by applications that are required to reduce latency. When handling Early Notification, note that the frame descriptor becomes invalid. Only the first buffer descriptor is valid if an Early Notification interrupt occurs.

16.4.3.8 Free Descriptor Area (FDA) Registers

Free Descriptor Area Base Register (FDA_Bas) 0x1C

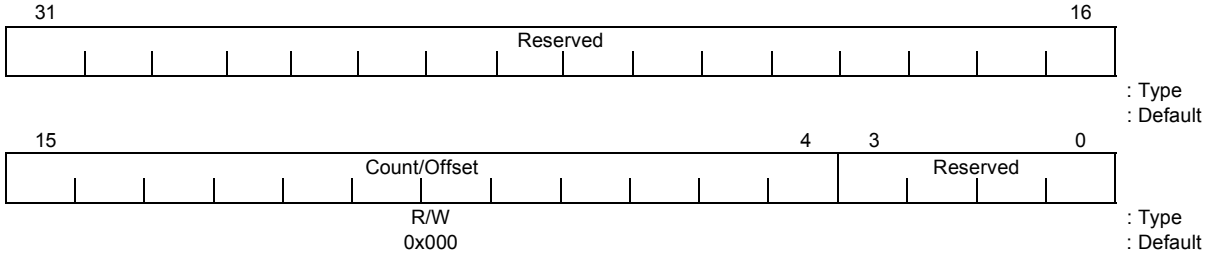


Bit(s)	Mnemonic	Field Name	Description
31:4		Address	Addr (Default: 0x000_0000, R/W) The Free Descriptor Area Base Register contains the start address of the area for writing the frame descriptors and buffer descriptors of reception packets. The address must be a multiple of 16 Bytes. Therefore, bits 0-3 are fixed to "0".
3:0		Reserved	

Figure 16.4.26 Free Descriptor Area Base Register

Hardware resets initialize the Free Descriptor Area Base Register to 0x0000_0000. Software resets do not change the register contents.

Free Descriptor Area Size Register (FDA_Lim) 0x20



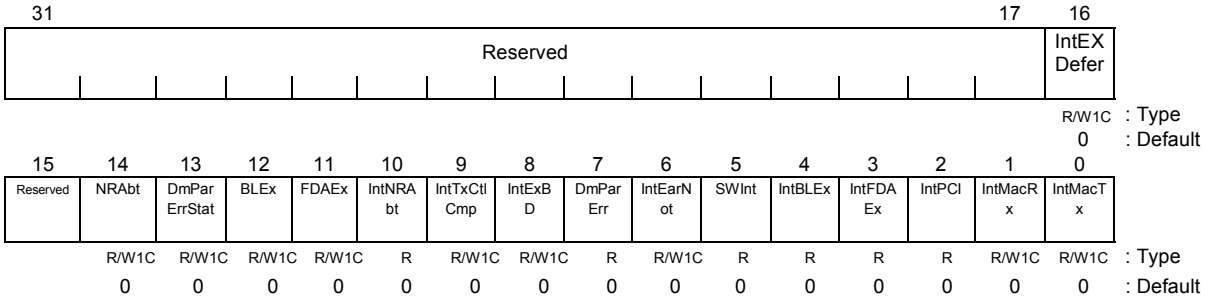
Bit(s)	Mnemonic	Field Name	Description
31:16		Reserved	
15:4	Count/Offset	Count/Offset	Count/Offset (Default: 0x000, R/W) This field sets the size of the Reception Descriptor Area in 16-Byte units. You could also consider the lower 16 bits as being an offset from the base address.
3:0		Reserved	

Figure 16.4.27 Free Descriptor Area Size Register

Hardware resets initialize the Free Descriptor Area Size Register to 0x0000_0000. Software resets do not change the register contents.

Note: You have to specify in the FDA_Lim Register the minimum offset value of the free descriptor area from which it is safe to start the next frame descriptor. You have to secure an area that is sufficient for storing a maximum size packet that includes one frame descriptor and the maximum number of buffer descriptors. For example, if the maximum number of buffer descriptors required to store a maximum length frame is 28, then this area requires a capacity of 256 (16 + 28 × 8 in 16-Byte units) Bytes.

16.4.3.9 Interrupt Source Register (Int_Src) 0x24



Bit(s)	Mnemonic	Field Name	Description
31:17		Reserved	
16	IntEXDefer		IntEXDefer (Default: 0, R/W1C) This bit is set if an excessive delay is detected and the EnExDefer (Excessive Defer Enable) bit of the Transmission Control Register (Tx_Ctl) is set.
15		Reserved	
14	NRAbt		NRAbt (Default: 0, R/W1C) This bit is set to "1" when a non-recoverable abort occurs.
13	DmParErrStat		DmParErrStat (Default: 0, R/W1C) This bit is set to "1" when a DMA parity error occurs.
12	BLEx		BLEx (Default: 0, R/W1C) This bit is set to "1' when the buffer list (BL) becomes full.
11	FDAEx		FDAEx (Default: 0, R/W1C) This bit is set to "1" when the free descriptor area (FDA) becomes full.
10	IntNRAbt		IntNRAbt (Default: 0, R) The interrupt source is a non-recoverable abort state.
9	IntTxCtlCmp		IntTxCtlCmp (Default: 0, R/W1C) The interrupt source is the completion of MAC control frame transmission.
8	IntExBD		IntExBD (Default: 0, R/W1C) The interrupt source is excessive buffer descriptors (more than 28 buffer descriptors).
7	DmParErr		DmParErr (Default: 0, R) The interrupt source is a DMA parity error.
6	IntEarNot		IntEarNot (Default: 0, R/W1C) The interrupt source is early notification.
5	SWInt		SWInt (Default: 0, R) The interrupt source is a software interrupt request.
4	IntBLEx		IntBLEx (Source: 0, R) The interrupt source is the buffer list being completely used up.
3	IntFDAEx		IntFDAEx (Default: 0, R) The interrupt source is the free descriptor area being completely used up.
2	IntPCI		IntPCI (Default: 0, R) The PCI Status Register (PCI_Stat) displays the interrupt source.

Figure 16.4.28 Interrupt Source Register (1/2)

Bit(s)	Mnemonic	Field Name	Description
1	IntMacRx		IntMacRx (Default: 0, R/W1C) The MAC Reception Status Register (Rx_Stat) displays the interrupt source.
0	IntMacTx		IntMacTx (Default: 0, R/W1C) The MAC Transmission Status Register (Tx_Stat) displays the interrupt source.

Figure 16.4.28 Interrupt Source Register (2/2)

Writing "1" to W1Clr clears it to "0". This bit denotes the bits that clear interrupts. Any writing of "0" to these bits is ignored. "R" indicates that a bit is Read Only. Either clearing the source that set these bits to "1" or resetting Ethernet control clears these bits to "0".

Software resets initialize the Interrupt Source Register to 0x0000_0000.

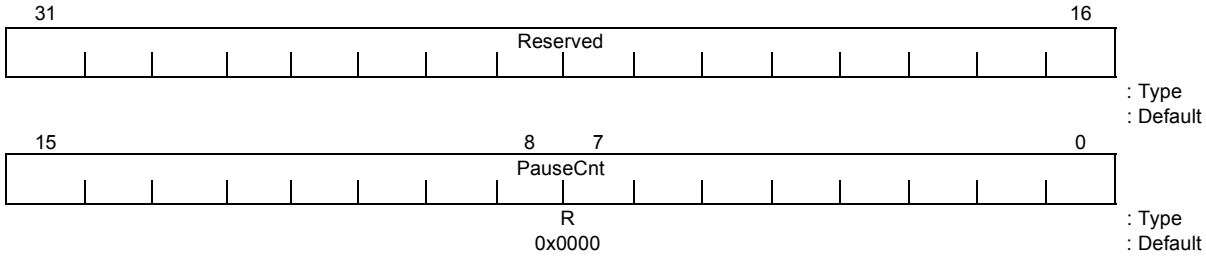
The system software reads the Interrupt Source Register to check whether there are any interrupts related to Ethernet control. Also, the Interrupt Source Register contains several status bits that can only be displayed here. If bits 15:0 are all "0", this indicates that the Ethernet Controller did not issue any interrupts. If the Ethernet Controller causes any interrupts, then setting the IntMask bit of the DMA Control Register makes it possible to mask any subsequent interrupts that the Ethernet Controller issues.

Bit 8 is set when there are more than 28 buffer descriptors are in a single frame descriptor. If excessive buffer descriptors occur, the controller aborts transmission or reception operation. To resume transmission or reception operation, you have to perform a software reset and rebuild the transmission queue, reception queue, and buffer list. Bit 13 is set when a parity error is detected in the DMA RAM. However, bit 7 is only set if the DParErrEn bit of the Interrupt Enable Register is set.

To make it possible to handle as many completed transmission frames or reception frames as possible, the software driver has to clear the IntMACTx bit or the IntMACRx bit. Since the frame that is the interrupt source is handled as a part of the immediately previous interrupt, there are cases where there are no complete frames in an interrupt. Take this into account when creating your software driver.

16.4.4 Flow Control Register group

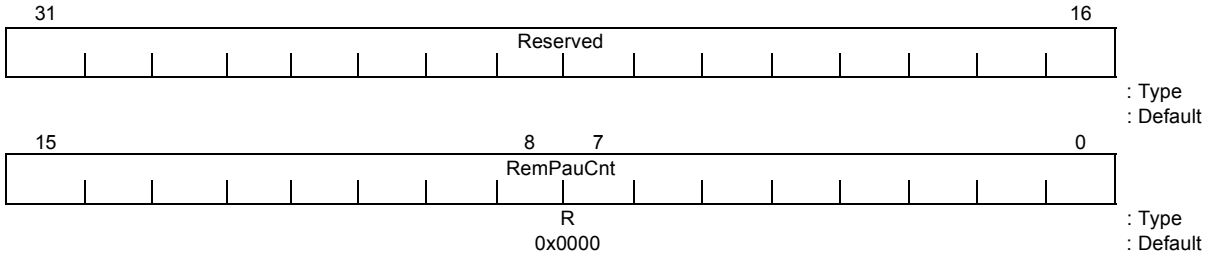
Pause Count Register (PauseCnt) 0x30



Bit(s)	Mnemonic	Field Name	Description
31:16		Reserved	
15:0	PauseCnt	Reception Pause Count	PauseCnt (Default: 0, R) This is the number of time slots that the transmitter pauses as a result of receiving a MAC control Pause operation packet.

Figure 16.4.29 Pause Count Register

Remote Pause Count Register (RemPauCnt) 0x34



Bit(s)	Mnemonic	Field Name	Description
31:16		Reserved	
15:0	RemPauCnt	Remote Pause Count	RemPauCnt (Default: 0, R) This is the number of time slots that the remote MAC pauses as a result of transmitting a Pause operation packet.

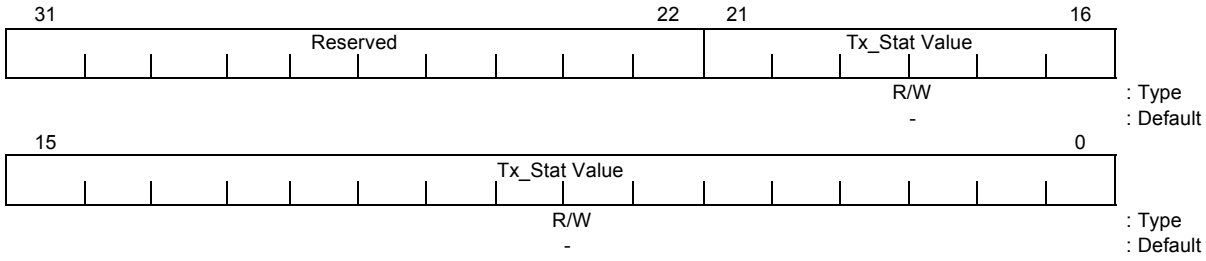
Figure 16.4.30 Remote Pause Count Register

Software resets initialize the Pause Count Register and Remote Pause Count Register to 0x0000_0000.

The Pause Count Register displays the current value of the received Pause count. When a Pause operation MAC control frame is received, the specified Pause count is set and the count is decremented at each unit time stamp. When the count is "0", it indicates that MAC is not pausing. The Remote Pause Count Register displays a rough value of the pause counter that the remote station has based on the transmitted Pause command.

In either case, the unit is 1 slot time. In other words, it is 512-bit late time.

Transmission Control Frame Status Register (TxCtlFrmStat) 0x38



Bit(s)	Mnemonic	Field Name	Description
31:22		Reserved	
21:0	Tx_Stat Value	Transmission Control Frame Status	Tx_Stat (Default: -, R/W) The Transmission Control Frame Status Register displays via the SdPause bit of the Transmission Control Register the status from when the MAC control frame was transmitted to a remote station.

Figure 16.4.31 Transmission Control Frame Status Register

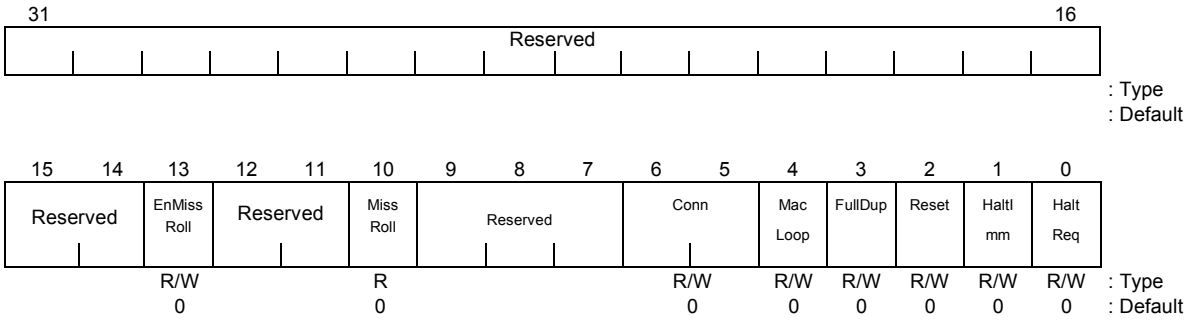
The Transmission Control Frame Status Register uses part of the internal RAM, so it is not affected by hardware resets or software resets.

The Transmission Control Frame Status Register displays via the SdPause bit of the Transmission Control Register the status when the MAC control frame was transmitted to a remote station. After transmission ends, the software driver fetches the status from this register. The software can reset this register before starting transmission of the MAC control frame. The software can use the TxCtlCmpEn bit of the Interrupt Enable Register to issue an interrupt when transmission of the MAC control frame ends.

For information on the bit fields, see the explanation of the Transmission Status Register (Tx_Stat) in 16.4.5.3 Transmission Control, Status Registers.

16.4.5 MAC Control, Status Register group

16.4.5.1 MAC Control Register (MAC_Ctl) 0x40



Bit(s)	Mnemonic	Field Name	Description
31:14		Reserved	
13	EnMissRoll	Missing Error Counter Rollover Enable	EnMissRoll (Default: 0, R/W) Issues an interrupt when the count value of the Missing Error Count Register rolls over from 0x7FFF to 0x8000.
12:11		Reserved	
10	MissRoll	Missing Error Counter Rollover	MissRoll (Default: 0, R) Indicates that the count value of the Missing Error Count Register rolled over from 0x7FFF to 0x8000. (Read only)
9:7		Reserved	
6:5	Conn	Connection Mode	Conn (Default: 00, R/W) This field selects the connection mode. 00: Automatic (default) 01: Reserved 10: MII (MII clock determines the transfer rate) 11: Reserved
4	MacLoop	MAC Loop Back	MacLoop (Default: 0, R/W) Directly provides the transmission signal as the input of the reception circuit without sending it outside the Ethernet Controller.
3	FullDup	Full Duplex Mode	FullDup (Default: 0, R/W) Set this bit to "1" for full duplex.
2	Reset	Software Reset	Reset (Default: 0, R/W) Resets all State Machines and FIFOs of the Ethernet Controller.
1	HaltImm	Immediate Halt	HaltImm (Default: 0, R/W) Immediately halts transmission or reception when set to “1”. When receiving data, if this bit is set after MAC starts processing the recipient address, reception operation for the current packet continues and the data is transferred to system memory. At this time, if the RxHalted bit of the Reception Status Register is set, it indicates that the system sent a Reception Halt Request while a packet was being received. If this bit is set before starting processing of the recipient address, reception operation is immediately halted. The RxHalted bit will then be set.
0	HaltReq	Halt Request	HaltReq (Default: 0, R/W) Halts transmission/reception when the packet currently in progress ends.

Figure 16.4.32 MAC Control Register

Hardware resets initialize the MAC Control Register to 0x8000. Setting the Reset bit (bit 2) executes software reset. Starting software reset clears bit 2. The other bits do not affect software resets.

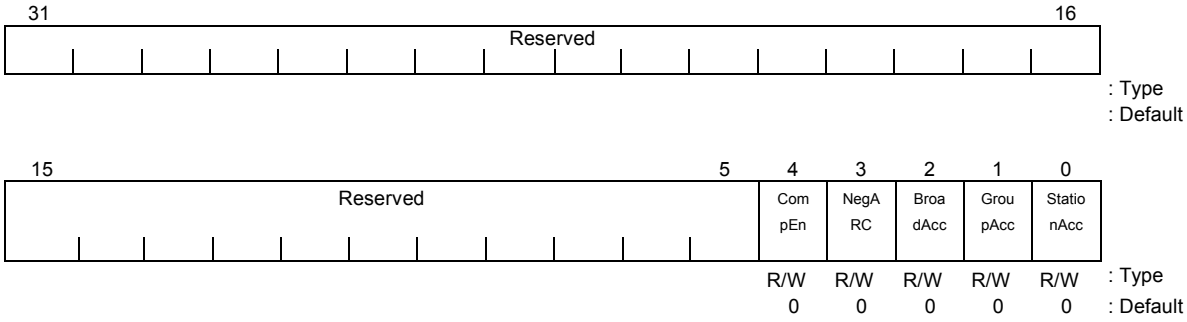
The MAC Control Register is used to display total control and status information of MAC. The MissRoll bit is the status bit. All other bits are control bits.

After the Reset bit is set, 4 MII transmission or reception clock cycles pass, then software reset is executed for several cycles. Therefore, after writing to the Reset bit, do not access the Ethernet Controller until 320 ns pass for 100 Mbps transfer or until 3,200 ns pass for 10 Mbps transfer. Before performing reset, you can use the MAC Transmission Control Register (Tx_Ctl) or MAC Reception Control Register (Rx_Ctl) to issue a Halt Request to end the current network transaction.

The MissRoll bit is set when the counter rolls over from 0x7FFF to 0x8000, and is reset when the software reads the Missing Error Count Register. See 16.4.5.7 Missing Error Count Register for an explanation.

Some PHYs do not support full duplex transfer. The MacLoop bit has higher priority than the FullDup bit.

16.4.5.2 ARC Control Register 0x44



Bit(s)	Mnemonic	Field Name	Description
31:5		Reserved	
4	CompEn	Compare Enable	CompEn (Default: 0, R/W) Enables the Compare mode.
3	NegARC	Negative ARC	NegARC (Default: 0, R/W) 0: Accepts packets the ARC recognized, but refuses all other packets. 1: Refuses packets the ARC recognized, but accepts all other packets.
2	BroadAcc	Broadcast Accept	BroadAcc (Default: 0, R/W) Accepts all packets that have a broadcast address.
1	GroupAcc	Group Accept	GroupAcc (Default: 0, R/W) Accepts all packets that have a multicast group address.
0	StationAcc	Station Accept	StationAcc (Default: 0, R/W) Accepts all packets that have a unicast station address.

Figure 16.4.33 ARC Control Register

Hardware resets initialize the ARC Control Register to 0x0000. Software resets do not change the register contents.

- The ARC recognizes the three following Ethernet address types.
- Station addresses: The first byte is even such as in 00-00-00-00-00-00.
- Broadcast addresses: Defined as FF-FF-FF-FF-FF-FF.

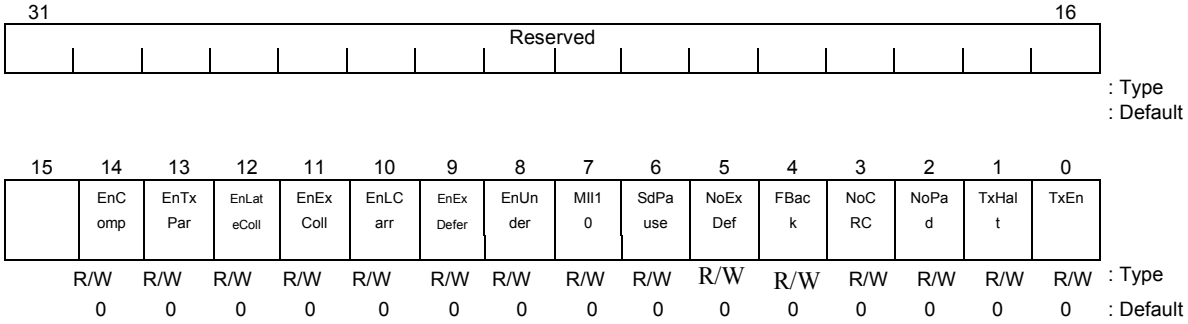
Multicast group address: The first byte is odd such as in 01-00-00-00-00-00. However, not FF-FF-FF-FF-FF-FF.

When the CompEn bit is set and the ARC's Compare mode is validated, the recipient address of the reception packet is compared with the address stored in ARC memory. For information on the structure of ARC memory, see 16.3.8.6 Address Recognition Circuit (ARC) operation. When the CompEn bit is cleared, the ARC unconditionally halts address comparisons. When an Accept bit (StationAcc, GroupAcc, or BroadAcc) is set, packets the ARC refused are also accepted. To refuse all packets, clear all bits of the ARC Control Register. To put MAC in the Promiscuous mode and accept all normal packets, set the ARC to accept all three of the above address types. You can also put MAC in the Promiscuous mode by setting the Negative ARC bit and clearing the CompEn bit.

When the ARC Compare mode is enabled, addresses for filtering reception messages are read from ARC memory. ARC memory has a 6-Byte structure for each entry. You can set it to Valid or Invalid for each entry as described in ARC Enable Register on page 86.

16.4.5.3 Transmission Control, Status Registers

Transmission Control Register (Tx_Ctl) 0x48


31
16
Reserved
: Type
: Default
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
EnComp EnTxPar EnLateColl EnExColl EnLCarr EnExDefer EnUnder MII10 SdPause NoExDef FBack NoCRC NoPad TxHalt TxEn
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W : Type
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 : Default

Bit(s)	Mnemonic	Field Name	Description
31:15		Reserved	
14	EnComp	Enable Complete	EnComp (Default: 0, R/W) Issues an interrupt either when MAC transmits or destroys 1 packet.
13	EnTxPar	Enable Transmission Parity	EnTxPar (Default: 0, R/W) Issues an interrupt when a parity error occurs in MAC Transmission FIFO.
12	EnLateColl	Enable Late Collisions	EnLateColl (Default: 0, R/W) Issues an interrupt if a collision occurs after more than 512-bit late time (64-Byte) time passes.
11	EnExColl	Enable Excessive Collisions	EnExColl (Default: 0, R/W) Issues an interrupt if collisions occur in the same packet 16 times.
10	EnLCarr	Enable Lost Carrier	EnLCarr (Default: 0, R/W) Issues an interrupt if either Carrier Sense could be not detected or was lost while transmitting a packet.
9	EnExDefer	Enable Excessive Defer	EnExDefer (Default: 0, R/W) Issues interrupts when MAC causes a MAX_DEFERRAL time delay. MAX_DEFERRAL = 0.24288 ms for 100 Mbps = 2.4288 ms for 10 Mbps
8	EnUnder	Underline Enable	EnUnder (Default: 0, R/W) Issues an interrupt when the MAC Transmission FIFO becomes empty during transmission.
7	MII10	MII 10 Mbps Mode	MII10 (Default: 0, R/W) Setting this bit to "1" enables SQE checking.
6	SdPause	PAUSE Transmission	SdPause (Default: 0, R/W) Transmits either the Pause command or another MAC control frame.
5	NoExDef	No Excessive Deferrals	NoExDef (Default: 0, R/W) Suppresses excessive deferral checking.
4	FBack	Fast Back Off	Fback (Default: 0, R/W) Uses a fast back-off timer during testing.
3	NoCRC	No Suppress	NoCRC (Default: 0, R/W) Does not add CRC to the end of a packet.
2	NoPad	No Padding	NoPad (Default: 0, R/W) Does not generate Pad Bytes even for packets with less than 64 Bytes.
1	TxHalt	Transmission Halt Request	TxHalt (Default: 0, R/W) Halts transmission if the current packet ends regardless of the packet type.
0	TxEn	Transmission Enable	TxEn (Default: 0, R/W) Immediately halts transmission when cleared to "0".

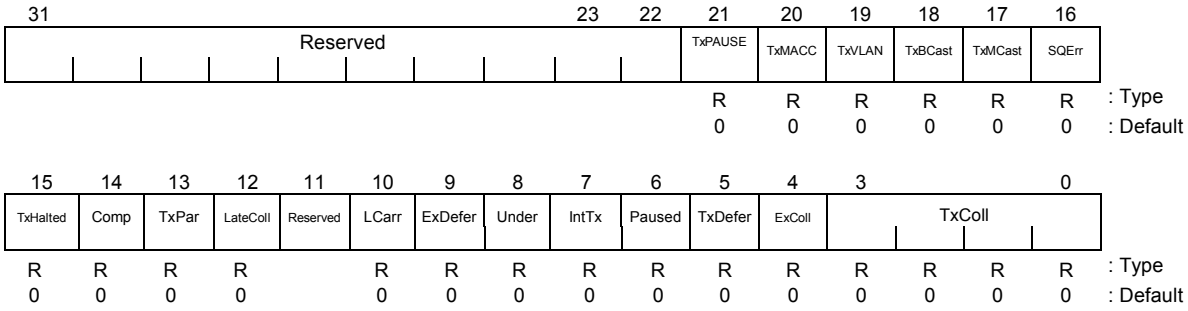
Figure 16.4.34 Transmission Control Register

A hardware reset initializes the Transmission Control register to 0000h. A software reset clears the TxEn and SdPause bits, but does not affect the other bits. Do not set the FBack (Fast Back-off) bit to "1" when in the normal operation mode.

The SdPause (Pause Transmission) bit is automatically cleared when transmission of the MAC control frame ends. Any writing of "0" to this bit is ignored.

To issue an interrupt for each packet, set the EnComp bit or all MAC Error Enable bits. You can also set interrupts to occur when a specific state occurs.

Transmission Status Register (Tx_Stat) 0x4C



Bit(s)	Mneminc	Field Name	Description
31:22		Reserved	
21	TxPAUSE	Pause Packet Communication	TxPAUSE (Default: 0, R) This bit is set when MAC transmits a MAC control Pause packet.
20	TxMACC	MAC Control Packet Transmission	TxMACC (Default: 0, R) This bit is set when MAC transmits a MAC control packet.
19	TxVLAN	VLAN Tag Packet Transmission	TxVLAN (Default: 0, R) This bit is set when MAC transmits a VLAN tag packet.
18	TxBCast	Broadcast Transmission	TxBCast (Default: 0, R) This bit is set when MAC transmits a broadcast packet.
17	TxMCast	Multicast Transmission	TxMCast (Default: 0, R) This bit is set when MAC transmits a multicast packet.
16	SQErr	Signal Quality Error	SQErr (Default: 0, R) Indicates that the Heartbeat signal could not be monitored at the end of a transmission.
15	TxHalted	Transmission Halted	TxHalted (Default: 0, R) Halts transmission by either clearing the TxEn bit of the Transmission Control Register or setting the HaltImm bit of the MAC Control Register.
14	Comp	Complete	Comp (Default: 0, R) Indicates that MAC either transmitted or destroyed one packet.
13	TxPar	Transmission Parity Error	TxPar (Default: 0, R) Indicates that a parity error was detected in MAC Transmission FIFO.
12	LateColl	Late Collision	LateColl (Default: 0, R) Indicates that 512-bit late time (64-Byte time) or more elapsed before a collision occurred.
11		Reserved	
10	LCarr	Lost Carrier	LCarr (Default: 0, R) Indicates that, during transmission, either Carrier Sense could not be detected or it was lost.
9	ExDefer	Excessive Deferral	ExDefer (Default: 0, R) Indicates that MAC deferred transmission for more than MAX_DEFERRAL.
8	Under	Underrun	Under (Default: 0 ,R) Indicates that MAC Transmission FIFO became empty during transmission.

Figure 16.4.35 Transmission Status Register (1/2)

Bit(s)	Mnemonic	Field Name	Description
7	IntTx	Transmission Interrupt	IntTx (Default: 0, R) This bit is set when interrupt conditions specified by the Transmission Control Register are met in a packet transmission.
6	Paused	Transmitter Paused	Paused (Default: 0, R) This bit is set if transmission is paused after the current packet ends.
5	TxDefer	Transmission Deferred	TxDefer (Default: 0, R) Indicates that a packet was kept waiting due to transmission delay.
4	ExColl	Excessive Collisions	ExColl (Default: 0, R) This bit is set if collisions occur 16 times in the same packet. Instead of transmitting that packet, processing of the next packet transmission starts.
3:0	TxColl	Transmission Collision Count	TxColl (Default: 0, R) Number of collisions that occurred when transmitting one packet.

Figure 16.4.35 Transmission Status Register (2/2)

Software resets initialize the Transmission Status Register to 0x00_0000. Also, this register is cleared at the beginning of each transmission packet.

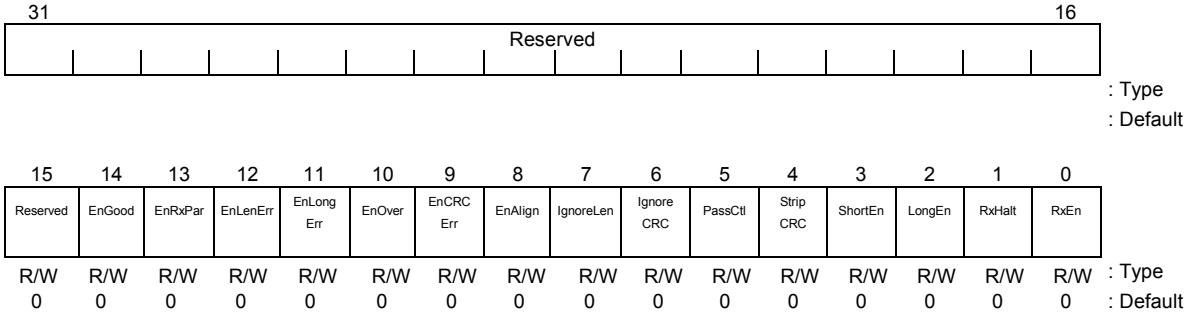
The Transmission Status flag is set each time the applicable event occurs. Also, an interrupt occurs if the corresponding bit of the Transmission Control Register is set.

The lower 5 bits of the Transmission Status Register indicate the collision count of the packet. In other words, when ExColl=1, TxColl becomes 0. If TxColl is not 0, then ExColl=0.

The MAX_DEFERRAL time is 0.24288 ms for 100 Mbps and 2.42880 ms for 10 Mbps. If the TxMCast bit (bit 17) and the TxBCast bit (bit 18) are both "0", they indicate that a unicast packet was transmitted.

16.4.5.4 Reception Control, Status Register

Reception Control Register (Rx_Ctl) 0x50



Bit(s)	Mnemonic	Field Name	Description
31:15		Reserved	
14	EnGood	Transmission Enable	EnGood (Default: 0, R/W) Issues an interrupt if a packet is received without any error.
13	EnRxPar	Enable Reception Parity	EnRxPar (Default: 0, R/W) Issues an interrupt if a parity error is detected in the MAC Reception FIFO.
12	EnLenErr	Enable Frame Length Errors	EnLenErr(Default: 0, R/W) Issues an interrupt if a Frame Length error is detected.
11	EnLongErr	Enable Long Errors	EnLongErr (Default: 0, R/W) If the LongEn bit is not set, an interrupt is issued when a frame longer than 1518 Bytes (1522 Bytes for VLAN) is received.
10	EnOver	Enable Overflows	EnOver (Default: 0, R/W) Issues an interrupt if the MAC Reception FIFO becomes full when receiving a packet.
9	EnCRCErr	Enable CRC Errors	EnCRCErr (Default: 0, R/W) Issues an interrupt either when CRC receives an invalid packet or PHY asserts Rx_er while receiving a packet.
8	EnAlign	Enable Alignment	EnAlign (Default: 0, R/W) Issues an interrupt if CRC receives an invalid packet with a length that is not a multiple of 8.
7	IgnoreLen	Ignore Frame Length	IgnoreLen (Default: 0, R/W) Does not check the frame length.
6	IgnoreCRC	Ignore CRC Value	IgnoreCRC (Default: 0, R/W) Does not check CRC.
5	PassCtl	Pass MAC Control Frame	PassCtl (Default: 0, R/W) Passes a received MAC control frame to the system.
4	StripCRC	Strip CRC Value	StripCRC (Default: 0, R/W) Checks CRC, but removes it from the message.
3	ShortEn	Enable Short	ShortEn (Default: 0, R/W) Enables the reception of frames shorter than 64 Bytes.[1]

Figure 16.4.36 Reception Control Register (1/2)

[1] The above frame length does not include a preamble and a Start Frame Delimiter (SFD). For details, see 16.3.8.1 Format of MAC frames and packets.

Bit(s)	Mnemonic	Field Name	Description
2	LongEn	Long Enable	LongEn (Default: 0, R/W) Enables the reception of frames longer than 1518 Bytes (1522 Bytes in the case of VLAN).[1]
1	RxHalt	Reception Halt Request	RxHalt (Default: 0, R/W) Halts reception after the current packet ends regardless of the packet type.
0	RxEn	Reception Enable	RxEn (Default: 0, R/W) This bit halts reception when cleared to "0". If this bit is cleared after MAC starts processing the recipient address, reception operation for the current packet continues and data is transferred to the system memory. At this time, the TxHalted bit of the Reception Status Register is set and indicates that the system issued a Reception Halt Request while receiving a packet. If this bit is cleared before processing of the recipient address starts, reception operation immediately halts. At this time, the RxHalted bit is set.

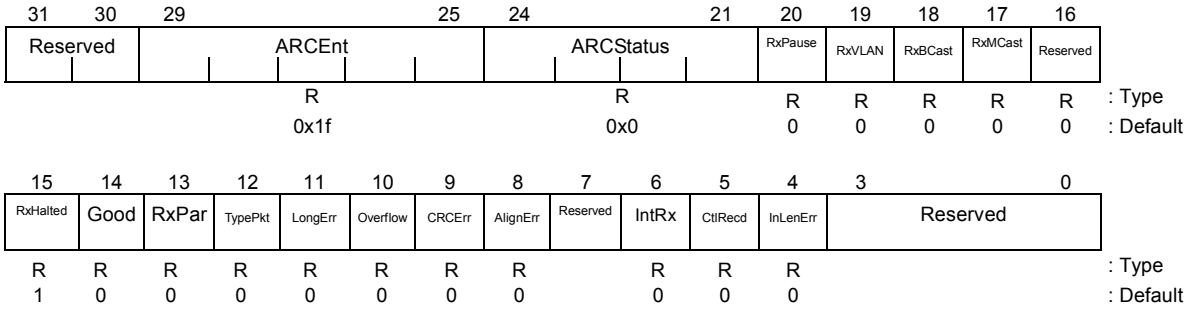
Figure 16.4.36 Reception Control Register (2/2)

Hardware resets initialize the Reception Control Register to 0x0000. Software resets clear the RxEn bit, but no other bits are changed.

To issue an interrupt for each packet, set the EnGood bit or all Error Enable bits. You can also set the Reception Control Register to only issue interrupts when a particular state occurs.

[1] The above frame length does not include a preamble and a Start Frame Delimiter (SFD). For details, see 16.3.8.1 Format of MAC frames and packets.

Reception Status Register (Rx_Stat) 0x54



Bit(s)	Mnemonic	Field Name	Description
31:30		Reserved	
29:25	ARCEnt	ARC Entry	ARCEnt (Default: 0x1F, R) This field stores the ARC entry index if the address matches. If the address does not match, all bits become "1".
24:21	ARCStatus	ARC Status	ARCStatus (Default: 0, R) Denotes ARC operation (see the following encoding).
20	RxPause	PAUSE Packet Reception	RxPause (Default: 0, R) This bit is set when the reception packet is a MAC control PAUSE packet.
19	RxVLAN	VLAN Tagged Packet Reception	RxVLAN (Default: 0, R) This bit is set when the reception packet is a VLAN tagged packet.
18	RxBCast	Broadcast Reception	RxBCast (Default: 0, R) This bit is set when the reception packet is a broadcast packet.
17	RxMCast	Multicast Reception	RxMCast (Default: 0, R) This bit is set when the reception packet is a multicast packet.
16		Reserved	
15	RxHalted	Reception Halted	RxHalted (Default: 1, R) This bit indicates that reception was halted either when the RxEn bit of the Reception Control Register was cleared or the HaltImm bit of the MAC Control Register was set.
14	Good	Normal Reception	Good (Default: 0, R) Indicates that a packet was received without any errors occurring.
13	RxPar	Reception Parity Error	RxPar (Default: 0, R) Indicates that a parity error was detected in the MAC Reception FIFO.
12	TypePkt	Type Packet	TypePkt (Default: 0, R) The value of the Frame Length field is greater than 1500 (no frame length check is performed). The software can use this bit to check the Protocol Type field.
11	LongErr	Long Error	LongErr (Default: 0, R) Indicates that a frame longer than 1518 Bytes (1522 Bytes for VLAN)[1] was received. However, if the LongEn bit of the Reception Control Register is set, this bit is not set.
10	Overflow	Overflow	Overflow (Default: 0, R) Indicates that the MAC Reception FIFO is full and a reception byte is missing.
9	CRCErr	CRC Error	CRCErr (Default: 0, R) Indicates that either the CRC value at the end of the packet does not match the calculated value or PHY asserted Rx_er while receiving a packet.

Figure 16.4.37 Reception Status Register (1/2)

[1] The above frame length does not include a preamble and a Start Frame Delimiter (SFD). For details, see 16.3.8.1 Format of MAC frames and packets.

Bit(s)	Mnemonic	Field Name	Description
8	AlignErr	Alignment Error	AlignErr (Default: 0, R) Indicates that the frame length is not an 8-bit multiple and that CRC is invalid.
7		Reserved	
6	IntRx	Reception Interrupt	IntRx (Default: 0, R) This bit is set when interrupt conditions are met due to the reception of a packet. If the EnGood bit of the Reception Control Register is set, notification is also sent for packets received without any errors. Note: If the Address Recognition Circuit (ARC) destroys a received packet when the EnGood bit or EnCRCErr bit of the Reception Control Register is set, this IntRx bit is also set if the Good or CRCErr bit of the Reception Status Register is set. In this case however, no interrupt notification is sent and the IntMacRx bit of the Int_Src Register is not set.
5	CtlRecd	Control Frame Received	CtlRecd (Default: 0, R) This bit is set if the ARC recognizes an address when the received packet is the MAC control frame (type=0x8808).
4	InLenErr	In Range Frame Length Error	InLenErr (Default: 0, R) If the value of the Frame Length field is 46 or less, this bit is set when the reception packet size is not 64 Bytes. Also, if the value of the Frame Length field is between 47 and 1500, this bit is set when the reception packet size is not the value of the Frame Length field + 18 (14 when in the Strip CRC mode).
3:0		Reserved	

Figure 16.4.37 Reception Status Register (2/2)

Software resets initialize the Reception Status Register to 0x3E00_8000. This register is also cleared at the beginning of each reception packet.

The Reception Status flag is set each time the applicable event occurs. Once the Reception Status flag is set, it remains set until the next packet is reached. When the corresponding bit of the Reception Control Register is set, an interrupt is issued.

The CtlRecd bit is set when the packet type is 0x8808 and the ARC recognizes an address. If both the RxMCast bit (bit 17) and the RxBCast bit (bit 18) are "0", they indicate that a unicast packet was received.

The ARCStatus field is encoded as follows below.

0000: Toss. MAC control frame received. PassCtl=0.

0001: Toss. The packet length is less than the minimum packet length, but is 6 Bytes or more. Keep if that is not the case.

0010: Toss. Matches ARC. NOT filtering.

0011: Reserved

0100: Toss. External CAM bit. NOT filtering.

0101: Reserved

0110: Toss. Address does not match. No external CAM. Compare disable. AND filtering.

0111: Toss. State in which the packet length is too short, the ARC result is Invalid, and the ShortEn bit is not set.

1000: Keep. Broadcast, multicast, or unicast accepting is enabled and the address matches.

1xx1: Reserved

1010: Keep. ARC matches. AND filtering.

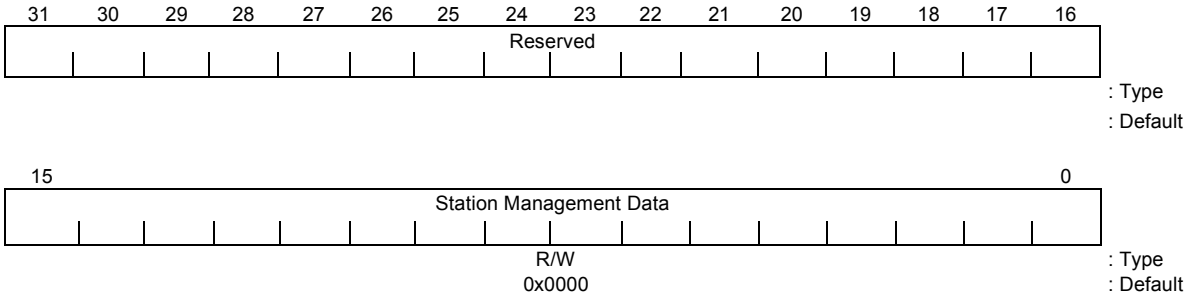
1100: Keep. External CAM bit. AND filtering.

1110: Keep. ARC does not match. AND filtering.

Note: The minimum packet length is 64 Bytes if ShortEn is not asserted, or is 14 Bytes if ShortEn is asserted.
Racing occurs between the internal ARC and external CAM. There are cases where the first signal to notify a hit blocks the other signal that notifies a hit and the ARC's Entry Status bit changes.

16.4.5.5 Station Management Registers

Station Management Data Register (MD_Data) 0x58

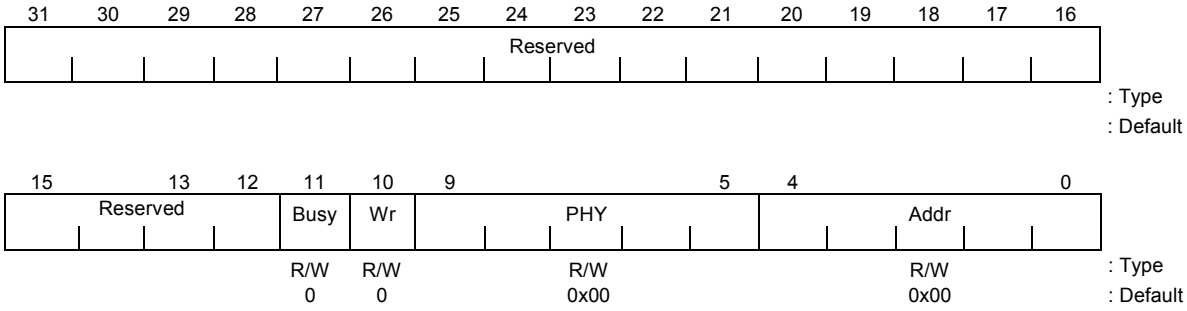


Bits	Mnemonic	Field Name	Description
31:16		Reserved	
15:0		Station Management Data	(Default: 0x0000, R/W) The MII item of the IEEE802.3 specification that relates the 100Base-T or 100 Mbps Ethernet defines the format of the Stage Management Data Register. For information on other hardware-dependent registers, see the separate PHY data sheet.

Figure 16.4.38 Station Management Data Register

Software resets initialize the Station Management Data Register to 0x0000.

Station Management Control, Address Register (MD_CA) 0x5C



Bit(s)	Mnemonic	Field Name	Description
31:12		Reserved	
11	Busy	Busy Bit	Busy (Default: 0, R/W) This bit is set when operation starts. The Ethernet Controller clears this bit when the operation ends.
10	Wr	Write	Wr (Default: 0, R/W) This bit is set to "1" during writes and is cleared to "0" during reads.
9:5	PHY	PHY Address	PHY (Default: 0, R/W) Address of all PHY to be read or written
4:0	Addr	Address	Addr (Default: 0, R/W) Address of PHY-internal register to be read or written.

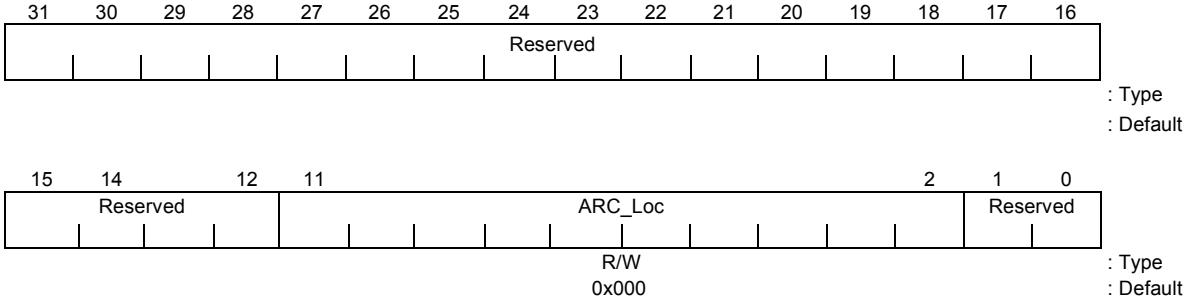
Figure 16.4.39 Station Management Control Address Register

Before accessing the PHY Control Register, use the software to check the Busy bit of the Station Management Control, Address Register and confirm that it is not set.

The Ethernet Controller supports reading or writing of Station Management data to PHY. The setting of the Station Management Control, Address Register does not affect the operation of the Ethernet Controller.

16.4.5.6 Address Recognition Circuit (ARC) Access Register

ARC Address Register (ARC_Adr) 0x60



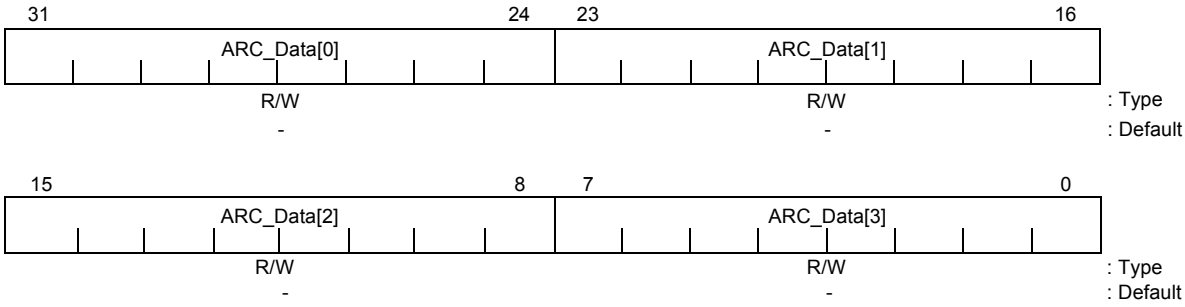
Bits	Mnemonic	Field Name	Description
31:12		Reserved	
11:2	ARC_Loc	ARC Address	ARC_Loc (Default: 0x00, R/W) ARC address: 4 bytes
1:0		Reserved	

Figure 16.4.40 ARC Address Register

Software resets initialize the ARC Address Register to 0x0000.

During normal operation, the ARC Address Register and ARC Data Register can perform read or write operation to all ARC areas including two double-word positions immediately after flow control ARC. (See Figure 16.3.12.) During normal operation, writes to all other memory positions are invalid. When the TestMode bit of the DMA Control Register is set, you can use the ARC Address Register and perform read or write operation on all RAM areas in the DMA Block.

ARC Data Register (ARC_Data) 0x64



Bits	Mnemonic	Field Name	Description
31:24	ARC_Data[0]	ARC Data Register [0]	ARC_Data[0] (Default -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.
23:16	ARC_Data[1]	ARC Data Register [1]	ARC_Data[1] (Default -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.
15:8	ARC_Data[2]	ARC Data Register [2]	ARC_Data[2] (Default: -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.
7:0	ARC_Data[3]	ARC Data Register [3]	ARC_Data[3] (Default: -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.

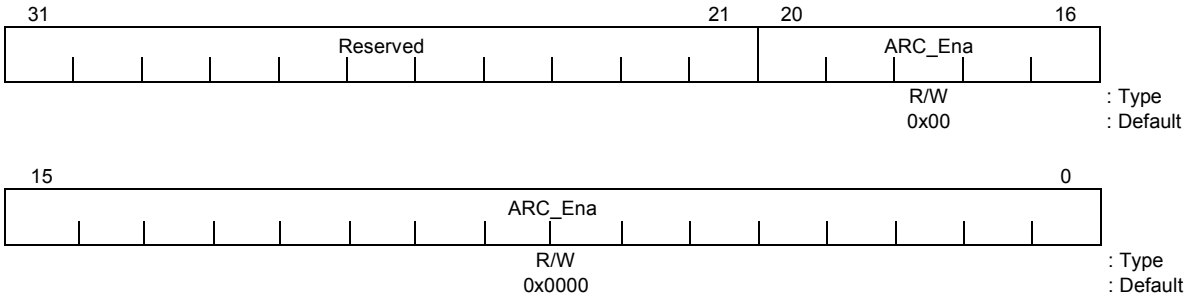
Figure 16.4.41 ARC Data Register

4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to. In other words, the PCI Byte Enable signal is ignored. When changing only 2 Bytes of a 4-Byte word, the system software must perform a Read/Modify/Write.

The ARC Data Register stores copies of data in ARC at the address specified by the ARC Address Register. You can read the content of the ARC Data Register as many times as you require. Writing data to this register changes the selected ARC data.

Note: In contrast to transmission/reception data in the Master mode, data transferred via the ARC Interface is interpreted in Big Endian.

ARC Enable Register (ARC _Ena) 0x68



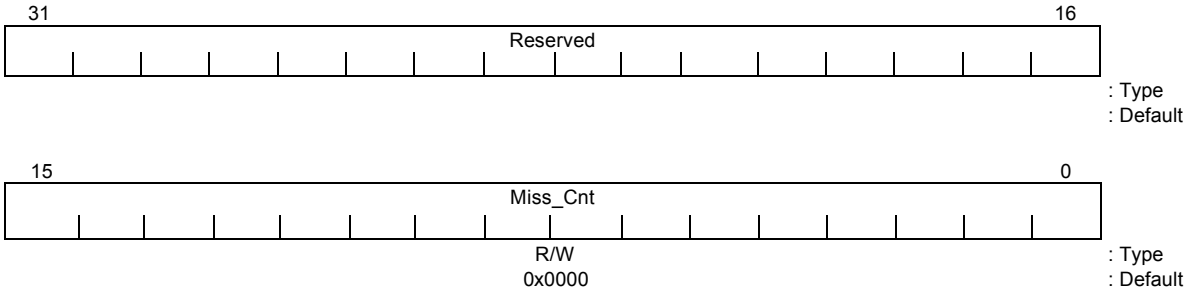
Bits	Mnemonic	Field Name	Description
31:21		Reserved	
20: 0	ARC_Ena	ARC Enable	ARC_Ena (Default: 0x0_0000, R/W) Sets the bits corresponding to an entry valid as address filtering.

Figure 16.4.42 ARC Enable Register

Hardware resets initialize the ARC Enable Register to 0x00_0000. Software resets do not change the contents of this register.

This register indicates which entry to validate as address filtering. This register can validate up to 21 entries numbered from 0 to 20.

16.4.5.7 Missed Error Count Register (Miss_Cnt) 0x7C



Bits	Mnemonic	Field Name	Description
31:16		Reserved	
15:0	Miss_Cnt	Missed Error Count	Miss_Cnt (Default: 0x0000, R/W) This register counts the number of valid packets MAC refused due to MAC Reception FIFO overflow, parity error, or the clearing of the Reception Enable bit (RxEn). This count does not include the packets refused by ARC.

Figure 16.4.43 Missed Error Count Register

Hardware resets initialize the Missed Error Count Register to 0x0000_0000. Software resets do not change the contents of this register.

The Missed Error Count Register displays the number of packets destroyed by various error types. This register provides the information required for station management along with the status information of the transmission/reception packet.

Performing read access to the Missed Error Count Register clears it. Therefore, the software is responsible for increasing the bit count and retaining the precise total error count.

The MissRoll bit of the MAC Control Register is set when the Missed Error Count Register rolls over from 0x7FFF to 0x8000. Also, in this case an interrupt occurs if the EnMissRoll bit is set.

To generate interrupts more frequently from the Station Management software, you can set the Missed Error Count Register to a value that is close to the final count value of 0x7FFF. For example, if you set this register to 0x7F00, an interrupt occurs when errors have occurred 256 times.

0x74 through 0x78 is reserved to maintain compatibility with previous Toshiba products.

17. SPI (Serial Peripheral Interface) Module (SPI)

17.1 Characteristics

The SPI is a serial interface that consists of clock, data output, and data input. The SPI is used to interface with serial power, serial A/D converters, other devices including simple serial clocks and data interfaces. The TX4938 only operates as a Master. It generates SPI clocks to Slaves. Multi-slave devices can share the SPI by using a unique Chip Select for each Slave device. Chip Select uses one general-purpose I/O port of the TX4938, or can generate other output ports that the system can use. If the Chip Select of a device is asserted and the device is selected, that device uses the SPICLK and SPIOUT signals to shift data in, and then uses the SPIIN signal to shift data out. If the device is not selected, the data output connected to SPIIN must be put into the tri-state and other devices must be able to share the SPIIN signal. The SPI module contains registers that can program the SPI CLK rate, MSB first or LSB first, clock polarity, data phase polarity, and Byte mode or Word mode operation.

The SPI module has the following characteristics:

- Selectable clock phase and polarity
- Transfer Size: 8-bit or 16-bit
- 4-frame Transmitter Buffer and 4-frame Receiver Buffer
- Master Operation
- Interframe Delay Time Counter
- MSB/LSB First

17.2 Block diagram

The SPI Module mainly consists of a 16-bit SPI Data Register (SPDR), a 16-bit Transmitter Buffer, a 16-bit Receiver Buffer, a 16-bit Shift Register, a Baud Rate Generator, an Interframe Delay Time counter, and interrupt logic. Figure 17.2.1 is a block diagram of the SPI Module.

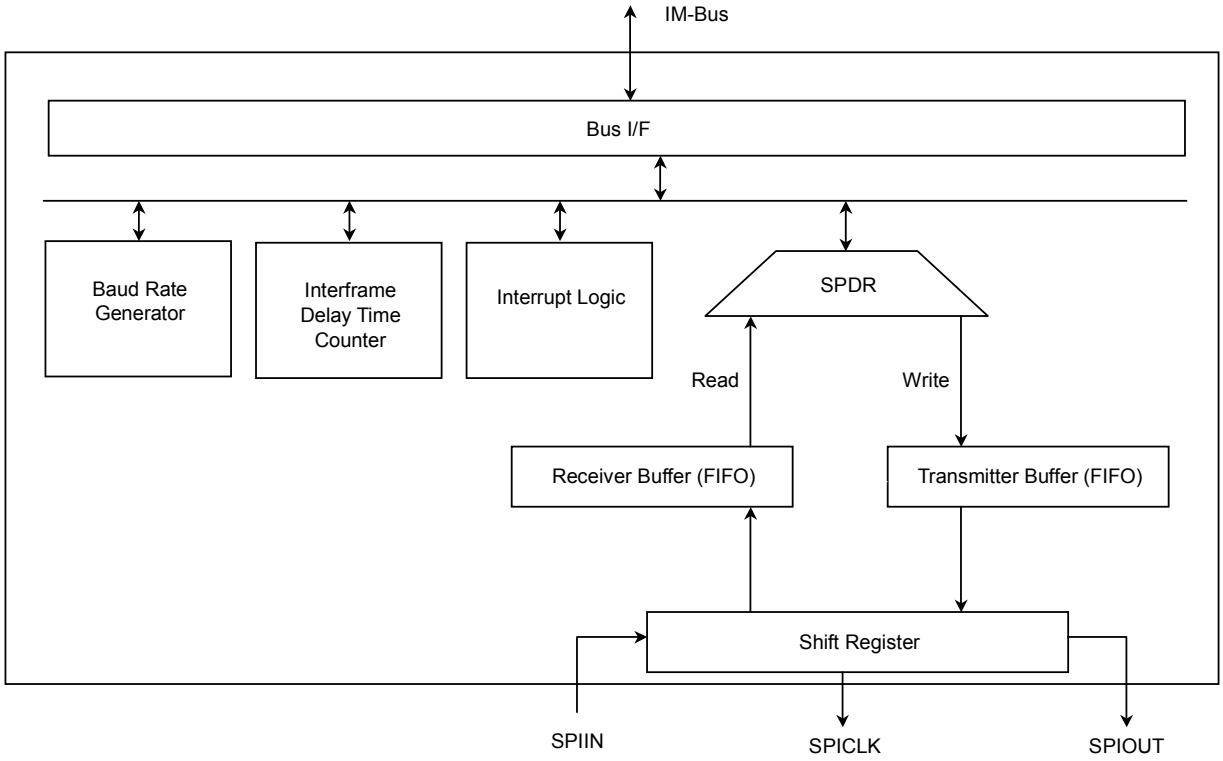


Figure 17.2.1 SPI Block Diagram

17.3 Operational description

17.3.1 Operation modes

The SPI Module has the two following operation modes:

- Configuration Mode (OPMODE = “01”):

  You can only rewrite the lower byte (bits[7:0]) of SPI Control Register 0 (SPCR0) and all bits of SPI Control Register 1 (SPCR1) when in this mode. Also, the SPSTP bit, Receiver FIFO and Transmitter FIFO are cleared and the SPI Module is reset when in this mode. Setting this mode forcibly terminates even the transfer of a frame that is currently in progress.

- Active Mode (OPMODE = “10”)

  The module operates in this mode during normal operation. You can execute transfers when in this mode.

17.3.2 Transmitter/Receiver

The SPI Module is in the Reset state when it is in the Configuration Mode. When in this mode, set the lower byte (bits[7:0]) of SPI Control Register 0 (SPCR0) and SPI Control Register 1 (SPCR1) to the desired value before changing the operation mode to the Active Mode. The SPI Module can start transferring data once it is in the Active Mode. Transfer starts when data is written to the SPI Data Register (SPDR).

The data written to the SPI Data Register (SPDR) is sent to the Shift Register, which then outputs the data to the Slave device. When the data is outputted from the SPIOUT pin, it is simultaneously fetched from the SPIIN pin. When fetching of the data is complete, the content of the Shift Register is loaded into the Receiver Buffer, the SRRDY bit of the SPI Status Register (SPSR) becomes “1”, then the Reception Buffer is notified that there is reception data. When the RBSI bit of the SPI Status Register (SPSR) is set to “1”, an interrupt occurs when the accumulated reception data reaches the level set by the RXIFL bit of SPI Control Register 0 (SPCR0).

When the content of the Transmitter Buffer is transferred to the Shift Register, the STRDY bit of the SPI Status Register (SPSR) becomes “1” and notification is sent that the Transmitter Buffer is available for use again. If the TBSI bit in the SPI Status Register (SPSR) is set to “1”, an interrupt is generated when the Transmitter Buffer has 1 to 4 locations emptied, as programmed in the TXIFL field in SPI Control Register 0 (SPCR0). Therefore, the software executes the following steps each time it writes data to the Transmitter Buffer.

(1) Check whether the STRDY bit or the TBSI bit is “1”. If neither bit is “1”, wait until one of them becomes “1”.
(2) Write data to the SPI Data Register (SPDR).

In this way, depending on the software used, the SPI Module can continue to seamlessly transmit data as long as the Transmitter Buffer is in the Run state until the data is shifted out from the Shift Register. If the software cannot keep up with the transfer rate, the SPI Module waits until the next data is written to the SPI Data Register (SPDR).

When one series of transmission is complete, the software deasserts the Chip Select signal of the Target device according to the following procedure.

(1) Check whether the SRRDY bit or the RBSI bit is "1". If neither bit is "1", wait until one of them is asserted.
(2) Check whether SIDLE is "1". If it is not "1", wait until it is asserted.
(3) Deassert the Chip Select signal.

The SPI Module supports either 8-bit or 16-bit per character operation as defined by the SSZ bit of SPI Control Register 1 (SPCR1). Also, the software can use the SBOS bit of SPI Control Register 0 (SPCR0) to select whether to shift MSB or LSB first. The other combination of control bits (SPHA and SPOL) determines the transfer format. See 17.3.4 for the transfer format.

17.3.3 Baud Rate Generator

The SPI Module operates IMBUSCLK as the Master Clock (SPI Master Clock). This module divides IMBUSCLK to generate (using the Baud Rate Generator) the SPICLK used for SPI transmission/reception. You can set this divide rate and can change the SPICLK to any frequency. Use SER[7:0] of SPI Control Register 1 (SPCR1) to set the divide rate. The following table shows the divide rate (SER[7:0]) and SPICLK frequency for when IMBUSCLK is 66 MHz or 60 MHz.

Table 17.3.1 SPICLK Frequency

SER[7:0]	SPI Clock Rate (IMBUSCLK: 66MHz) (TMPR4938XBG-333)	SPI Clock Rate (IMBUSCLK: 60 MHz) (TMPR4938XBG-300)
00000001b	16.5 MHz	15 MHz
00000010b	11 MHz	10 MHz
00000011b	8.25 MHz	7.5 MHz
00000100b	6.6 MHz	6 MHz
00000101b	5.5 MHz	5 MHz
…		
00001001b	3.3 MHz	3 MHz
…		
00010011b	1.65 MHz	1.5 MHz
…		
11111111b	128.91 KHz	117.19 KHz

17.3.4 Transfer format

During SPI transfer, serial data transmission (shift out) and reception (shift in) is executed simultaneously. At this time, serial data is shifted or sampled synchronous to the serial clock.

The SPHA bit and SPOL bit of SPI Control Register 0 (SPCR0) determine the transfer format. The setting of the SPHA bit broadly classifies the protocol that is used. Each protocol is described below in items 17.3.4.1 and 17.3.4.2.

17.3.4.1 SPHA =0 format

Figure 17.3.1 shows the transfer format when SPHA is “0”.

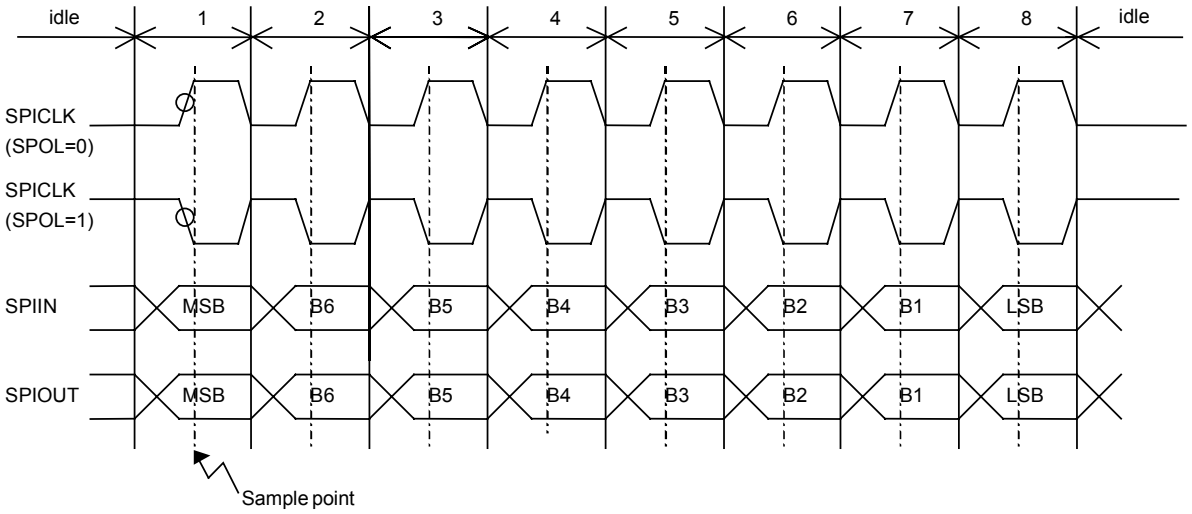


Figure 17.3.1 Transfer Format when SPHA is “0”

When in this format, data is sequentially fetched from the first clock edge after the Idle state. When the SPOL bit is “0”, data is fetched at the rising edge. When this bit is “1”, data is fetched at the falling edge. The SPIIN or SPIOUT signal is switched at the second clock edge of SPICLK. The signal is switched at the SPICLK falling edge when the SPOL bit is “0”, or is switched at the SPICLK rising edge when the SPOL bit is “1”. When the SPOL bit is “0”, SPICLK is at the Low level during the Idle state. When the SPOL bit is “1”, SPICLK is at the High level during the Idle state.

17.3.4.2 SPHA = 1 format

Figure 17.3.2 shows the transfer format when SPHA is "1".

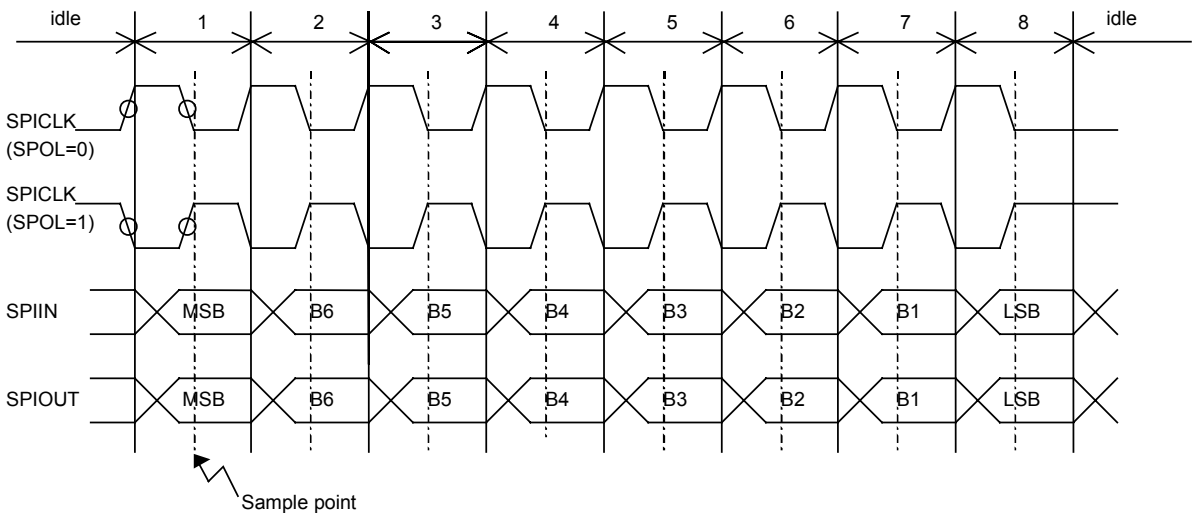


Figure 17.3.2 Transfer Format when SPHA is "1"

When in this format, data is sequentially fetched from the second clock edge after the Idle state. When the SPOL bit is "0", the data is fetched from the second falling edge. When the SPOL bit is "1", the data is fetched from the second rising edge. When the SPOL bit is "0", SPICLK is at the Low level during the Idle state. When the SPOL bit is "1", SPICLK is at the High level during the Idle state.

17.3.5 Interframe Delay Time Counter

There are cases where it is preferable to shorten the time between data groups. In such cases, the Interframe Delay Time Counter is used to specify the delay time between data groups. When 16 bits is selected as the data size by SPI Control Register 1, the delay time is inserted after 16-bit data is shifted. When 8 bits is selected, the delay time is inserted after 8-bit data is shifted. When IFS bits [9:0] are set to a value other than "0", a delay is inserted between the characters. The length of the delay inserted between characters is also changed by the IFSPSE bit of SPI Control Register 0 (SPCR0) as shown below.

When SPCR0.IFSPSE=1: SPI Master Clock Cycle × IFS[9:0] × 32

When SPCR0.IFSPSE=0: SPI Master Clock Cycle × IFS[9:0]

If the IFS value is "0", seamless operation is performed. The SPI Module continues to shift data and supply a clock as long as the software does not fall behind the transfer rate of the transmitter.

17.3.6 Buffer configuration

The SPI Module has a Transmitter Buffer and a Receiver Buffer. The buffers use FIFOs to store data. Each FIFO can store 4-frame data.

The Transmitter Buffer stores the value written to the SPI Data Register. Either when in the Idle state or when the current transmission is complete, the first data written to the Transmitter Buffer is transferred to the Shift Register.

On the other hand, data received in the Shift Register is stored in the Receiver Buffer each time transfer ends.

You can issue interrupts each time data equal in size to the data in the buffer accumulates.

17.3.7 SPI system errors

The SPI Module signals the following system errors during transfer.

17.3.7.1 Overrun error (SPOE)

An overrun error is issued when an attempt is made to write the next data to the Transmitter Buffer regardless of whether the Transmitter Buffer is full. The data to be newly written at this time is not written to the Transmitter Buffer. Also, the SPOE bit of the SPSR Register becomes "1".

17.3.8 Interrupts

The SPI Module has three types of interrupt sources. The result of OR operation performed on three interrupt sources is inputted to the Interrupt Controller (IRC) as SPI interrupts. Check the SPI Status Register (SPSR) to see which interrupts occurred.

Type	Status bits	Mask-able bit
System error or idle	SPOE, SIDLE	SOEIE, SILIE
Receive Buffer Fill	RBSI	RBSIE
Transmit Buffer Fill	TBSI	TBSIE

Use System errors or Idle interrupts for error detection or Idle state interrupts. Use Receiver Buffer Fill interrupts and Transmitter Buffer Fill interrupts when setting new transmission data to the buffer or when reading receiver data from the buffer.

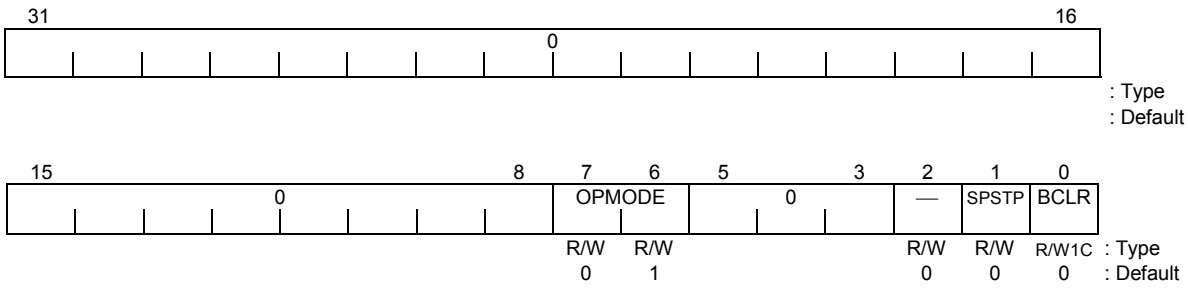
17.4 Registers

Access SPI Module registers using 32-bit access. Operation is not guaranteed when accessing these registers using any other access size. Do not write “0” to any undefined bits.

Table 17.4.1 SPI Module Registers

Offset Address	Bit Width	Register Symbol	Register Name
0xF800	32	SPMCR	SPI Master Control Register
0xF804	32	SPCR0	SPI Control Register 0
0xF808	32	SPCR1	SPI Control Register 1
0xF80C	32	SPFS	SPI Interframe Delay Time Register
0xF810	32	—	(Reserved)
0xF814	32	SPSR	SPI Status Register
0xF818	32	SPDR	SPI Data Register
0xF81C	32	—	(Reserved)

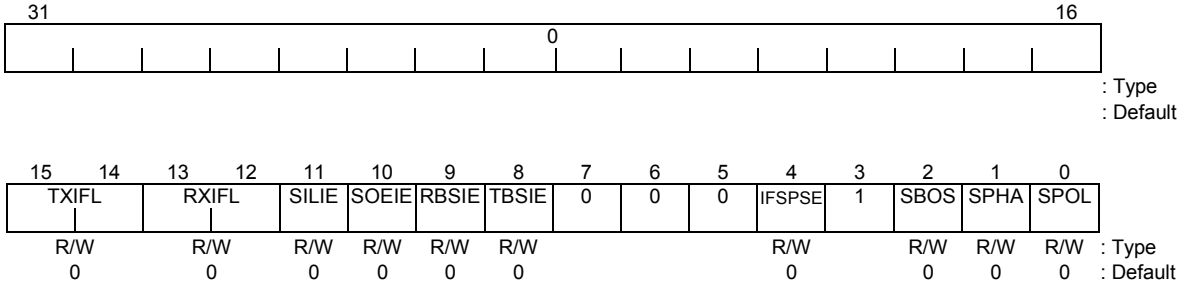
17.4.1 SPI Master Control Register (SPMCR) 0xF800



Bit(s)	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:6	OPMODE	Operation Mode	Operation Mode (Default: 00) Sets the operation mode. 00: Don’t care. 01: Configuration Mode 10: Active Mode 11: Reserved
5:3	—	Reserved	—
2	—	Reserved	Do not write “1” to this bit.
1	SPSTP	SPI Stop	SPI Stop (Default: 0) When “1” is written to this bit, the SPI Module does not transfer any more data after the current frame is complete. You can only set this bit when in the Active Mode. Entering the Configuration Mode clears this bit. 0: Normal operation 1: Halt after the current transfer ends.
0	BCLR	SPI Buffer Clear	SPI Buffer Clear (Default: 0) Use this bit to clear the Transmitter FIFO and Receiver FIFO. Writing “1” to this bit initializes the FIFO. Write “1” to this bit after the SPI Module enters the Idle state (SIDLE=1). Always outputs “0” when read. Write: 0: Don’t care 1:Clear the FIFO

Figure 17.4.1 SPI Master Control Register (SPMCR)

17.4.2 SPI Control Register 0 (SPCR0) 0xF804


31
16
0
: Type
: Default
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
TXIFL RXIFL SILIE SOEIE RBSIE TBSIE 0 0 0 IFSPSE 1 SBOS SPHA SPOL
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
0 0 0 0 0 0 0 0 0 0
: Type
: Default

Bit(s)	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:14	TXIFL	Transmit Interrupt Fill Level	Transmit Interrupt Fill Level (Default: 00) You can issue interrupts according to the number of free data in the Transmitter FIFO. This field sets the free data count of the Transmitter FIFO at which interrupts are issued. 00: Issue an interrupt when there is 1 or more free data in the Transmitter FIFO. 01: Issue an interrupt when there are 2 or more free data in the Transmitter FIFO. 10: Issue an interrupt when there are 3 or more free data in the Transmitter FIFO. 11: Issue an interrupt when there are 4 free data in the Transmitter FIFO.
13:12	RXIFL	Receive Interrupt Fill Level	Receive Interrupt Fill Level (Default: 00) You can issue interrupts according to the number of data in the Receiver FIFO. This field sets the number of Receiver FIFO data at which to issue an interrupt. 00: Issue an interrupt when there is 1 or more free data in the Receiver FIFO. 01: Issue an interrupt when there are 2 or more free data in the Receiver FIFO. 10: Issue an interrupt when there are 3 or more free data in the Receiver FIFO. 11: Issue an interrupt when there are 4 free data in the Receiver FIFO.
11	SILIE	SPI IDLE Interrupt Enable	SPI IDLE Interrupt Enable (Default: 0) Enables SPI Idle interrupts. 0: Disable 1: Enable
10	SOEIE	SPI Overrun Interrupt Enable	SPI IDLE Overrun Enable (Default: 0) Enables SPI Overrun interrupts. 0: Disable 1: Enable
9	RBSIE	Receive Buffer Fill Interrupt Enable	Receive Buffer Fill Interrupt Enable (Default: 0) This bit specifies whether to signal an interrupt to the Interrupt Controller based on the number of data actually in the Receiver FIFO or to only display it as a status. 0: Disable (mask) 1: Enable
8	TBSIE	Transmit Buffer Fill Interrupt Enable	Transmit Buffer Fill Interrupt Enable (Default: 0) This bit specifies whether to signal an interrupt to the Interrupt Controller based on the number of free data actually in the Transmitter FIFO or to only display it as a status. 0: Disable (mask) 1: Enable
7:5	—	Reserved	—

Figure 17.4.2 SPI Control Register 0 (SPCR0) (1/2)

Bit(s)	Mnemonic	Field Name	Description
4	IFSPSE	Inter Frame Space prescaler enable	Inter Frame Space prescaler Enable (Default: 0) Enables prescaler of the Interframe Delay Time Counter. 0: Disable (× 1) 1: Enable (× 32)
3	—	Reserved	—
2	SBOS	SPI Bit Order Select	SPI Bit Order Select (Default: 0) This bit specifies the bit order of the transfer data. 0: LSB first (transfer starting from the least significant bit) 1: MSB first (transfer starting from the most significant bit)
1	SPHA	SPI Phase	SPI Clock Phase (Default: 0) Selects the clock phase. 0: Samples at the first clock edge, then shifts at the second edge. 1: Shifts at the first clock edge, then samples at the second edge.
0	SPOL	SPI Polarity	SPI Clock Polarity (Default: 0) Selects the SPICLK polarity. 0: High Active (SPICLK is Low when idle) 1: Low Active (SPICLK is High when idle)

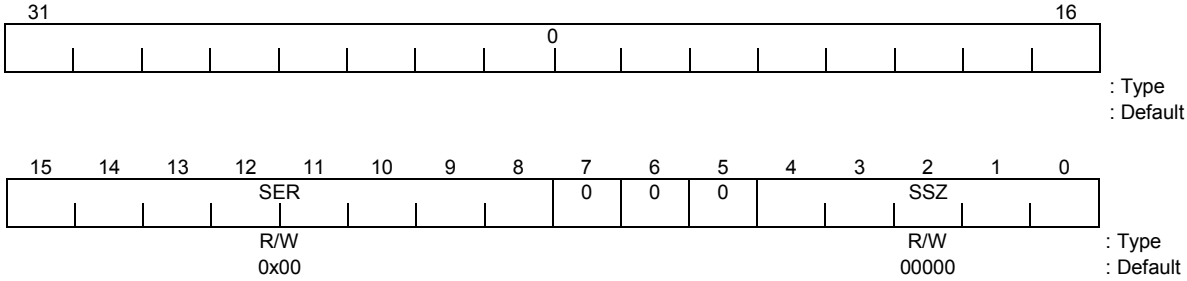
Note 1: You can only write to bits 4, 2, 1, or 0 when the SPI Module is in the Configuration Mode.

Note 2: The SPOL and SPHA bits select the SPICLK phase and the clock edge at which to sample data. For details, see 17.3.4 Transfer format.

Note 3: The value of the SPOL bit must be changed, with the device's chip select signal deasserted. (For example, the chip select signal can be controlled, using a general-purpose port of the TX4938.)

Figure 17.4.2 SPI Control Register 0 (SPCR0) (2/2)

17.4.3 SPI Control Register 1 (SPCR1) 0xF808

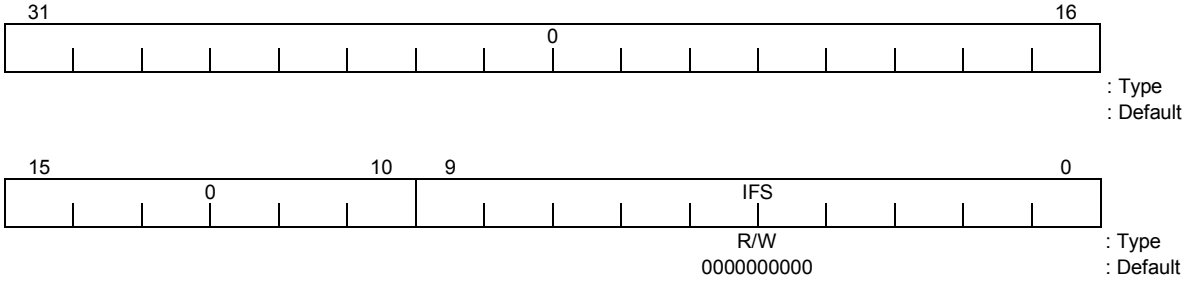


Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:8	SER	SPI Data Rate	SPI Data Rate (Default: 000000b) This field sets the transfer bit rate. The transfer bit rate is calculated according to the following equation. $f_{BR} = f_{SPI}/2 (n + 1)$ f_{BR} : SPICLK Frequency f_{SPI} : SPI $_{Master}$ Clock Frequency n : SER ($_{Setting}$ “0” is not permitted) (See 17.3.3 for SER and clock frequency examples.)
7:5	—	Reserved	—
4:0	SSZ	SPI Transfer Size	SPI Transfer Size (Default: 00000b) Selects the transfer size. 0x08: 8 bits 0x10: 16 bits Other values: Reserved (not settings are permitted)

Note 1: You can only write to this register when the SPI Module is in the Configuration Mode.

Figure 17.4.3 SPI Control Register 1 (SPCR1)

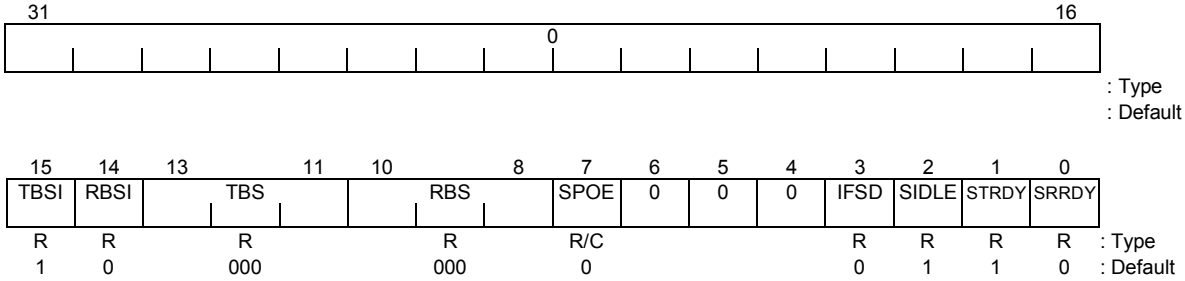
17.4.4 SPI Interframe Delay Time Counter (SPFS) 0xF80C



Bits	Mnemonic	Field Name	Description
31:10	—	Reserved	—
9:0	IFS	Inter Frame Space	Inter Frame Space (Default: 0x00) This register sets the delay time to insert between two consecutive frames. When this register is set to "0", transfer ends, the next transmission data is loaded from the Transmitter FIFO to the Transmitter Buffer, and then the next transfer is performed. (Since the time data is loaded into the buffer is not "0", the delay time from the last clock until the first clock of the next transfer is not actually "0".) When setting the IFSPSE bit of the SPCR0 Register to "0" and not using prescaler, the Interframe delay time is the value calculated using the formula in 17.3.5. (16.7 ns - 17.05 μs when SPI Master Clock Frequency is 60 MHz) When setting the IFSPSE bit of the SPCR0 Register to "1" and using prescalar, the Interframe delay time is the value calculated using the formula in 17.3.5. (533 ns - 545.6 μs when SPI Master Clock Frequency is 60 MHz) Only write to this register when the SIDLE bit is "0".

Figure 17.4.4 SPI Interframe Delay Time Counter (SPFS)

17.4.5 SPI Status Register (SPSR) 0xF814


31
16
0
: Type
: Default
15 14 13 11 10 8 7 6 5 4 3 2 1 0
TBSI RBSI TBS RBS SPOE 0 0 0 IFSD SIDLE STRDY SRRDY
R R R R R/C R R R R : Type
1 0 000 000 0 0 1 1 0 : Default

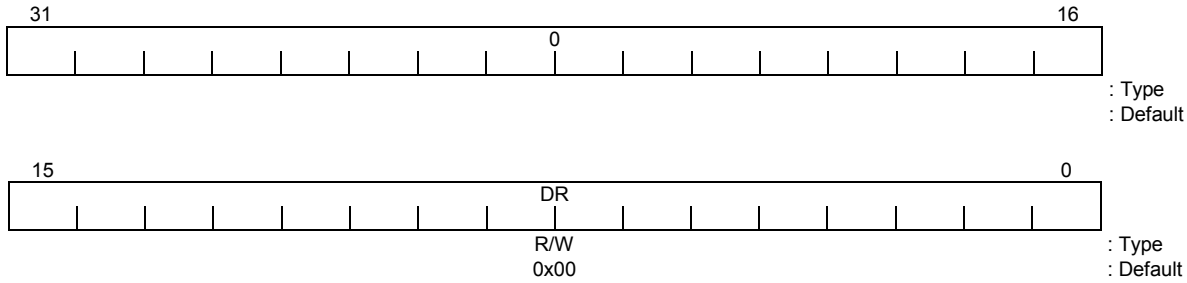
Bit(s)	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	TBSI	Transmit Buffer Status Indicator	Transmit Buffer Status Indicator (Default: 1) Indicates the status of interrupts according to the number of free data in the Transmitter FIFO. 0: No interrupt 1: Interrupt
14	RBSI	Receive Buffer Status Indicator	Receive Buffer Status Indicator (Default: 0) Indicates the status of interrupts according to the number of free data in the Receiver FIFO. 0: No interrupts 1: Interrupts
13:11	TBS	Transmit Buffer Status	Transmit Buffer Status (Default: 000) Indicates how many data are in the Transmit FIFO. 000: No data 001: 1 010: 2 011: 3 100: 4 (FIFO is full) 101 – 111: NA
10:8	RBS	Receive Buffer Status	Receive Buffer Status (Default: 000) Indicates how many data are in the Receiver FIFO. 000: No data 001: 1 010: 2 011: 3 100: 4 (FIFO is full) 101 – 111: NA
7	SPOE	SPI Overrun Error	SPI Overrun Error (Default: 0) Indicates that an overrun occurred in the Transmitter FIFO. You can clear this bit by writing "1" to it. You can also clear this flag by putting the SPI Module in the Configuration Mode. Read: 0: No errors 1: Generate overrun errors Write: 0: Don't care 1: Clear
6:4	—	Reserved	—

Figure 17.4.5 SPI Status Register (SPSR) (1/2)

Bit(s)	Mnemonic	Field Name	Description
3	IFSD	SPI Inter Frame Space Delay Indicator	SPI Inter Frame Space Delay Indicator (Default: 0) This bit becomes "1" when transfer of a frame ends and transfer of the next frame is deferred by the Interframe Delay Time Counter. 0: No interframe cycle 1: Interframe cycle
2	SIDLE	SPI Idle Indicator	SPI Idle Indicator (Default: 0) This bit becomes "1" either when the Transmitter FIFO is empty or the SPSTP bit is "1" and there is no transfer in progress. 0: Run 1: Idle
1	STRDY	SPI Transmit Ready	SPI Transmit Ready (Default: 0) Indicates that there is space in the Transmitter FIFO. 0: Transmitter FIFO is full. 1: Transmitter FIFO has space.
0	SRRDY	SPI Receive Ready	SPI Receive Ready (Default: 0) Indicates that there is reception data in the Receiver FIFO. This bit is cleared when the SPDR Register is read and there is no longer any valid data in the Receiver FIFO. 0: Receiver FIFO is empty. 1:There is data in the Receiver FIFO.

Figure 17.4.5 SPI Status Register (SPSR) (2/2)

17.4.6 SPI Data Register (SPDR) 0xF818



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:0	DR	SPI Data Register	SPI Data Register (Default: 0x00) The data written to this register is stored in the Transfer Buffer. When the Shift Register is empty, data is written to the Shift Register and transmission begins. You can read the Receiver FIFO data when you read this register. When the transfer data size is 8 bits, use the lower 8 bits of this register. When reading, the upper side (bits[15:8]) are "0".

Figure 17.4.6 SPI Data Register (SPDR)

18. NAND Flash Memory IPL

18.1 Features

NAND Flash Memory Initial Program Loader (NAND IPL), which is coded in the TX4938 on-chip ROM, enables to boot TX4938 from the NAND flash memory. The user writes NAND IPL control information and a user boot program to the NAND flash memory in the prescribed format. NAND IPL loads the user boot program from the NAND flash memory into RAM and continues the standard boot process.

- Specifying NAND IPL in the boot configuration boot memory starts up NAND IPL.
- The IPL control information includes the memory controller initialization sequence, the address or the size of the user boot program to be loaded. NAND IPL operates based on NAND IPL control information.
- NAND IPL supports rewriteable NAND flash memory to be connected to the NAND Flash Memory Controller, Mask ROM type NAND flash memory, and SmartMedia™, but not support the NAND flash memory with partially different configurations.
- NAND IPL requires a 4 Kbyte stack (work area). Since RAM performs initialization using the NAND IPL control information, you can use various types of RAM.
- The user may use any memory area in the NAND flash memory except the zone used with NAND IPL, and the area used with NAND IPL when it does not affect operation of NAND IPL. For details on the data specification, see 18.3.5.
- NAND IPL operates in either Big-endian or Little-endian mode. It also operates in Cache On or Write Through mode.
- NAND IPL uses PIO [0] bit 1, General Purpose Register ($4) or the lower address of External Bus Controller Channel 0 to signal the NAND IPL operation results.
- NAND IPL can not be modified since it is written into on-chip ROM.

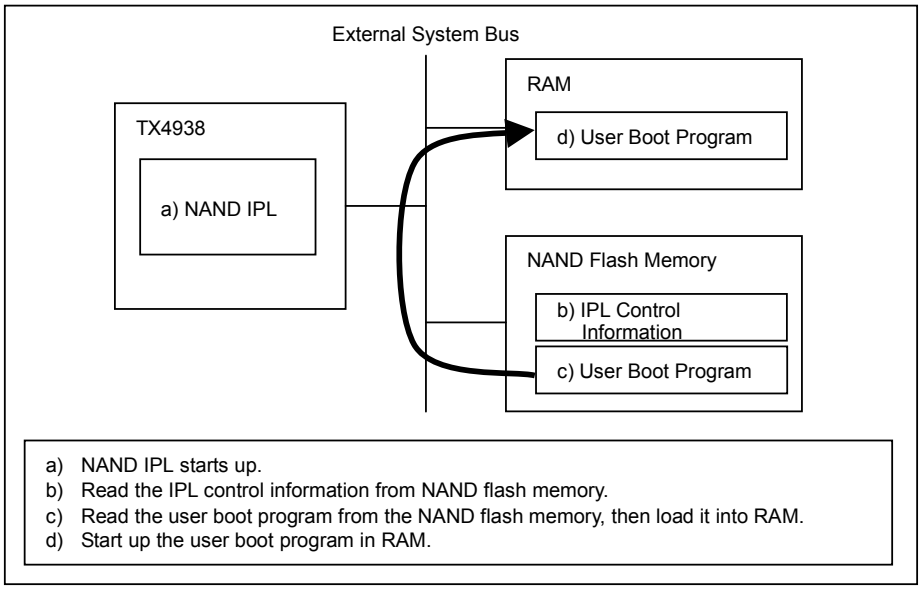

External System Bus
RAM
d) User Boot Program
TX4938
a) NAND IPL
NAND Flash Memory
b) IPL Control Information
c) User Boot Program
a) NAND IPL starts up.
b) Read the IPL control information from NAND flash memory.
c) Read the user boot program from the NAND flash memory, then load it into RAM.
d) Start up the user boot program in RAM.

Figure 18.1.1 Image Diagram of Booting from NAND Flash Memory

18.2 Block Diagram

18.2.1 System Block Diagram

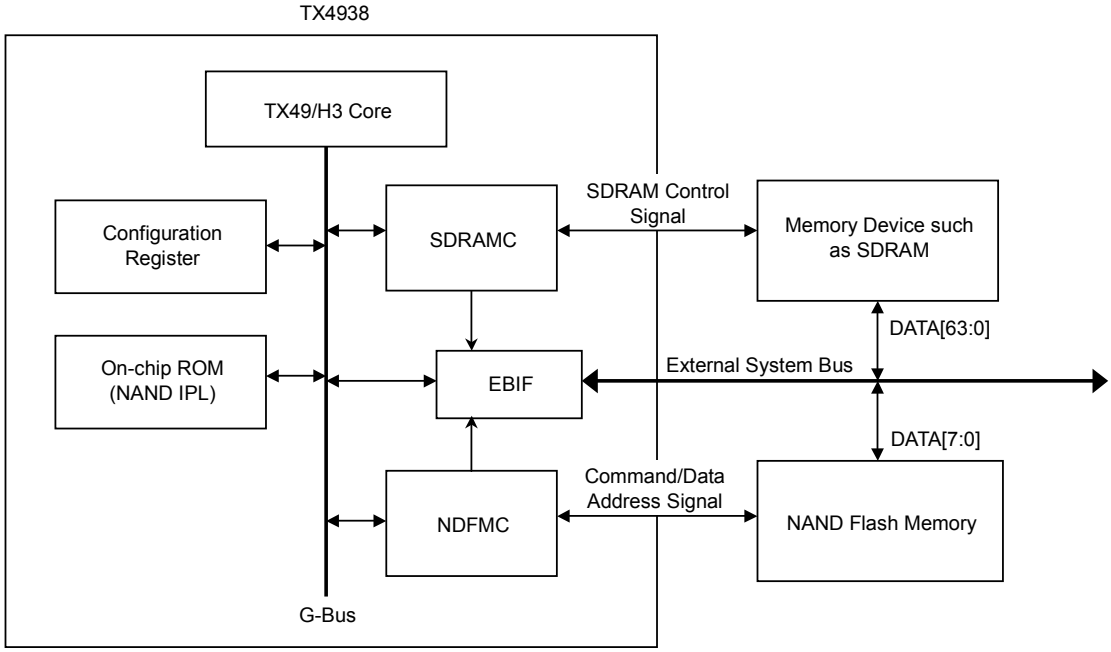


Figure 18.2.1 Block Diagram of System with NAND Flash Memory Connected to SDRAM

18.2.2 Function Block Diagram

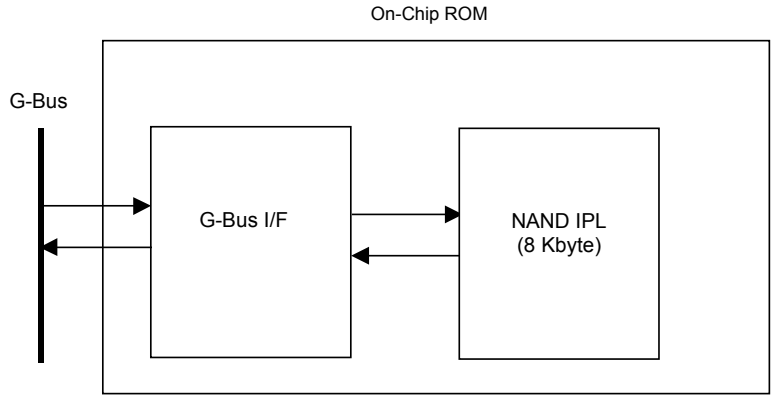


Figure 18.2.2 Function Block Diagram of NAND IPL

18.3 Detailed Explanation

18.3.1 NAND IPL Operation Conditions

(1) NAND IPL supports the Bi-endian (Big- and Little-endian) format. When the TX4938 operates in the Big-endian mode, the Big-endian user boot program may be loaded. When operating in the Little-endian mode, the Little-endian user boot programs may be loaded.

(2) The NAND flash memory to be used with NAND IPL is shown below. For the format of the user data to be written in NAND flash memory and data configuration of NAND flash memory, see 18.3.4 and 18.3.5. NAND flash memory must be readable on power up.

Table 18.3.1 Usable NAND Flash Memory

Size (MB)	Device ID		Bytes/Page	Pages/Block	Physical Block Count	Zone Count
	Flash ROM	Mask ROM				
8	0xE6	0xD6	512 + 16	16	1024	1
16	0x73	0x57		32		
32	0x75	0x58			2048	2
64	0x76	0xD9			4096	4
128	0x79	0xDA			8192	8

(3) NAND IPL uses PIO [0] and General Purpose Register $4 ($a0) to signal the NAND IPL operation results, and external bus controller channel for unusual conditions.

(4) Operating NAND IPL requires the 4 Kbyte stack for work area, and RAM having the same size as the user program in which the user boot program is stored. For the address map, see 18.3.3. For initial setting of RAM access and address setting of the work area and user boot program, see 18.3.4. As in the case of booting from the NOR flash memory, the user program requires the work area after jumping to the user boot program.

18.3.2 NAND IPL Operation

NAND IPL operates according to the following flow.

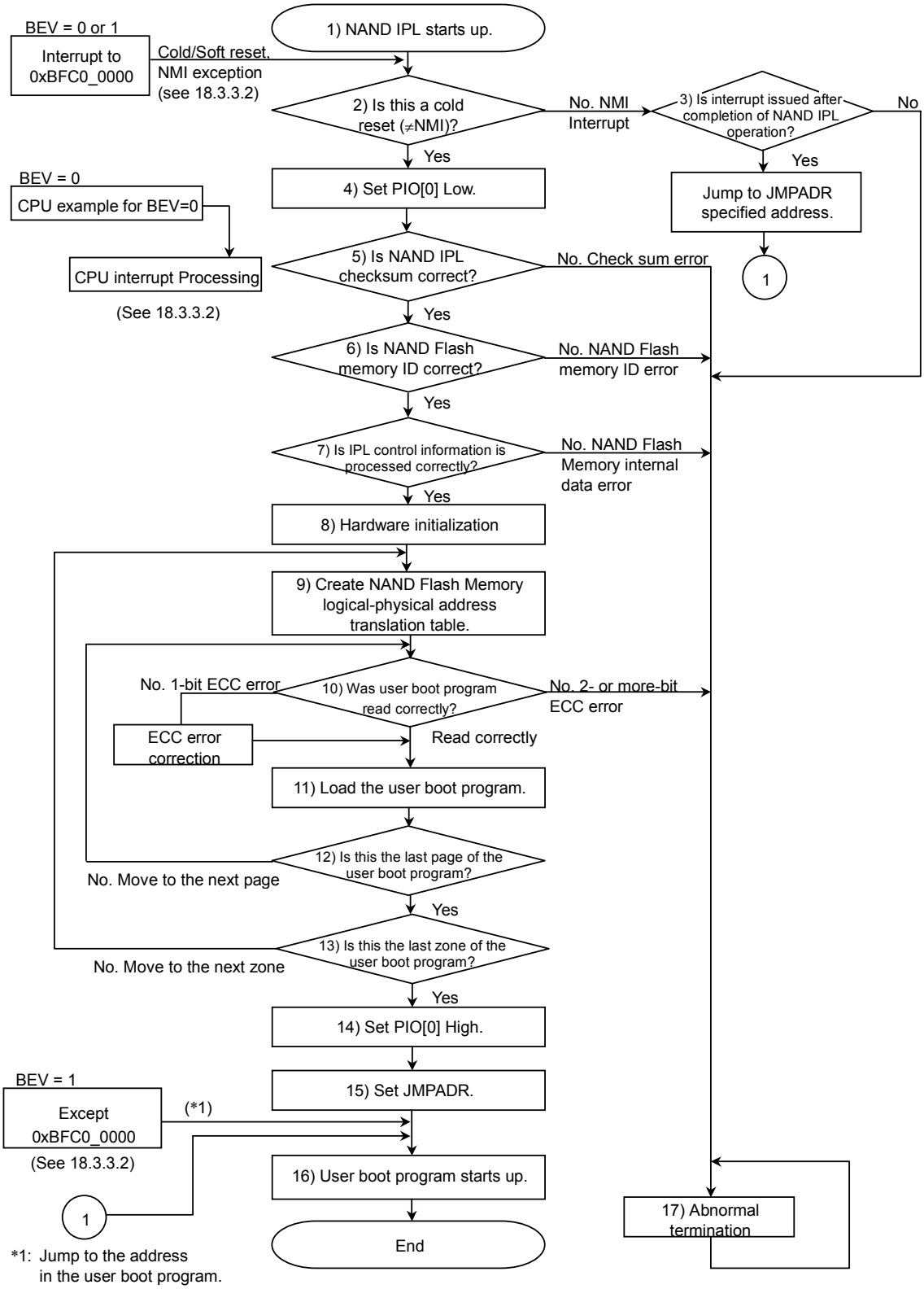

BEV = 0 or 1
Interrupt to 0xBFC0_0000
Cold/Soft reset, NMI exception (see 18.3.3.2)
1) NAND IPL starts up.
2) Is this a cold reset (≠NMI)?
No. NMI Interrupt
3) Is interrupt issued after completion of NAND IPL operation?
No
Yes
Jump to JMPADR specified address.
1
Yes
BEV = 0
CPU example for BEV=0
CPU interrupt Processing
(See 18.3.3.2)
4) Set PIO[0] Low.
5) Is NAND IPL checksum correct?
No. Check sum error
Yes
6) Is NAND Flash memory ID correct?
No. NAND Flash memory ID error
Yes
7) Is IPL control information is processed correctly?
No. NAND Flash Memory internal data error
Yes
8) Hardware initialization
9) Create NAND Flash Memory logical-physical address translation table.
No. 1-bit ECC error
10) Was user boot program read correctly?
No. 2- or more-bit ECC error
ECC error correction
Read correctly
11) Load the user boot program.
12) Is this the last page of the user boot program?
No. Move to the next page
Yes
13) Is this the last zone of the user boot program?
No. Move to the next zone
Yes
14) Set PIO[0] High.
BEV = 1
Except 0xBFC0_0000
(See 18.3.3.2)
(*1)
15) Set JMPADR.
16) User boot program starts up.
1
End
17) Abnormal termination
*1: Jump to the address in the user boot program.

Figure 18.3.1 NAND IPL Program Flow

18.3.2.1 NAND IPL Startup

To boot the TX4938, operation usually starts from the 0xBFC0_0000 program (user boot program) connected to External Bus Controller Channel 0. When you specify NAND IPL using Boot Configuration (see section 3.2 Boot Configuration for details), the on-chip ROM is allocated to 0xBFC0_0000 to operate NAND IPL. For details on the NAND IPL address map, see subsection 18.3.3.

18.3.2.2 Cold Reset

When cold reset or NMI interrupt occurs, TX49/H3 core transfers the control to 0xBFC0_0000 in NAND IPL. NAND IPL checks whether it is NMI interrupt by referring the SR bit in the Status register. When it is not the NMI interrupt, NAND IPL jumps to kseg0 and goes to the stage 4) Set PIO[0] Low in Figure 18.3.1.

Note: NAND IPL refers only the SR bit.

18.3.2.3 MNI Exception

When NAND IPL processing is completed, the NAND IPL interrupt handler jumps to the address set in the Jump Address register (JMPADR). When NAND IPL processing is not completed, it jumps to the abnormal processing routine in NAND IPL. For abnormal processing, see 18.3.6.

The JMPADR value is used to check whether NAND IPL processing is completed. The initial value 0xBFC0_0000 indicates that NAND IPL processing is not completed, and other values indicate the completion of NAND IPL processing.

Note: NAND IPL reads the JMPADR value, which is defined in the IPL control information defined by the user, and sets it in JMPADR. The user must not operate JMPADR.

18.3.2.4 Setting PIO[0] Low

NAND IPL uses bit 1 of PIO [0] to signal the program operation status to an external device. When NAND IPL operates, bit 1 of PIO [0] becomes Low.

18.3.2.5 NAND IPL Checksum

NAND IPL checks whether the NAND IPL code in on-chip ROM is correct.

18.3.2.6 NAND Flash Memory ID Check

NAND IPL checks whether the NAND flash memory being used can respond properly to NAND IPL by reading the NAND flash memory ID.

18.3.2.7 NAND IPL Control Information Processing

NAND IPL reads the data defined in compliance with the prescribed format from NAND flash memory. Since NAND IPL can not use RAM at this stage, access to the NAND flash memory by specifying a physical address. You can not use Error Check and Correct (ECC). Therefore, NAND IPL judges the correctness of the IPL control information using the following method. For more information on the prescribed format, see 18.3.4.

- The IPL control information consists of the same data set written three times consecutively. NAND IPL compares these three data sets on read of them, and regards as a correct one when at least two of the three data sets are identical.

NAND IPL performs the following operation to search the IPL control information.

- Performs normal block check of the NAND flash memory only with the data contained in the top page of the block "Block Status Area" (redundant area). NAND IPL does not perform this check on the applicable data in other pages.
- Since the NAND flash memory may contain bad blocks, you can not fix the IPL control information to a specific physical address. NAND IPL sequentially reads physical blocks 0 through 23, and searches each page within a block to find the IPL control information start ID on the top of the page when the block is good one.

18.3.2.8 Hardware Initialization

NAND IPL secures a stack (work area) which is used for RAM access setup (initialization) or NAND IPL in accordance to the IPL control information defined in the NAND flash memory.

Note: The external bus controller channel can be set by writing IPL control information, but that is not recommended due to processing speed. We recommend to perform the minimum hardware setting in NAND IPL such as NAND flash memory and RAM access required to operate IPL, and to use the boot program for other settings.

18.3.2.9 NAND Flash Memory Logical-physical Address Translation Table Creation

NAND IPL creates a logical-physical address translation table since NAND flash memory after RAM is secured is accessed using logical addresses. For the logical address, see 18.3.5.1.

NAND IPL creates the address translation table for every zone, and recreates it (*1) if the zone is changed.

(*1): NAND IPL has the information for only one zone, which indicates relationship between the logical-physical address translation table (translation table) and the target zone. Upon load of the user boot program (program), NAND IPL creates the translation table of the zone which resides in the top block of the program, and remembers the zone number before load of the program in this zone. If a program spans multiple zones, NAND IPL recreates the translation table of the new target zone when the block to be loaded by NAND IPL exits from the target zone, and loads the program of the new zone.

18.3.2.10 Checking Read of the User Boot Program

NAND IPL reads the user boot program from NAND flash memory in accordance to the information such as the address defined in the IPL control information. If a 1-bit error is detected upon read of data for one page, NAND IPL performs 1-bit ECC correction (see 18.3.5.1), when 2- or more-bit error is detected, NAND IPL jumps to abnormal termination processing. Note that you can not write this data back to the NAND flash memory after correcting 1-bit error on it.

18.3.2.11 Loading the User Boot Program

NAND IPL loads the user boot program for one page, which is read from the NAND flash memory, into RAM. As described in 18.3.2.10, NAND IPL corrects the user boot program if 1-bit error is detected, and loads it into RAM.

18.3.2.12 Checking the Last Page of the User Boot Program

NAND IPL checks whether the page being loaded is the last page in which the user boot program resides. If it is not the last page, NAND IPL returns to processing of 18.3.2.10 and continues load of the user boot program.

18.3.2.13 Checking the Last Zone of the User Boot Program

NAND IPL checks whether the zone being loaded with NAND IPL is the last zone in which the user boot program resides. If it is not the last zone, NAND IPL returns to processing of 18.3.2.9 and recreates the logical-physical address translation table before load of the user boot program.

18.3.2.14 Setting PIO[0] High

After loading is terminated normally, NAND IPL sets bit 1 of PIO [0] to High. PIO [0] which remains Low indicates that an error occurred and loading of the user boot program was not completed. For details, see 18.3.6 NAND Flash Memory IPL Error Handling.

18.3.2.15 Setting Jump Address Register (JMPADR)

Before transferring the control to the user boot program, NAND IPL rewrites the initial Jumper Address Register (JMPADR) value of 0xBFC0_0000 to the top address of the user boot program. For information on this register, see 5.2.8 Jump Address Register.

18.3.2.16 User Boot Program Startup

NAND IPL jumps to the value set with the jump address register (JMPADR), then starts up the user boot program and transfers the control to it.

The user needs to write processing for occurrence of CPU exception on the top address of the user boot program. For details, see 18.3.3.2.

18.3.2.17 Abnormal Termination

NAND IPL can detect multiple errors. For more information, see 18.3.6 NAND Flash Memory IPL Error Handling.

18.3.3 NAND Flash Memory IPL Address Map

18.3.3.1 Address Information When NAND IPL Starts up

When NAND IPL is specified by boot memory setting of Boot Configuration, NAND IPL is allocated to 0xBFC0_0000 which is the reset vector.

The area of the user boot program to be loaded by NAND IPL is either kseg0 or kseg1. You can set the load destination address to any address within this range. The user boot program is usually allocated to 0xBFC0_0000 in the TX49/H3 core mounted products. When NAND IPL is used, NAND IPL creates (loads) the user boot program at a RAM address.

18.3.3.2 Address Information When CPU Exception Is Issued

(1) Cold Reset/NMI Exception

If an exception occurs after startup of the user boot program and BEV=0 or BEV=1 is set to the TX49/H3 core Status Register, TX49/H3 core transfers the control to address 0xBFC0_0000.

NAND IPL checks whether an exception is an NMI exception. When it is an NMI exception, NAND IPL refers to the Jump Address Register (JPMADR) and transfers the control to the (JMPADR) address (*1), which is the top address of the user boot program. When it is not an NMI exception, execution starts from address 0xBFC0_0000. Therefore, you need to describe the exception vector for an NMI exception in the JMPADR address of the user boot program.

(*1): The jump destination address set in the Jump Address Register (JMPADR)

(2) TLB Refill Exception

If an exception occurs after startup of the user boot program and BEV=0 is set to the TX49/H3 core Status Register, the TX49/H3 core transfers the control to address 0x8000_0000.

When BEV=1, NAND IPL Exception Handler refers to the Jump Address Register (JMPADR) from exception vector address 0xBFC0_0200 and transfers the control to the (JMPADR) + 0x0200 address of the user boot program. Therefore, you need to describe an exception vector to the (JMPADR) + 0x0200 address of the user boot program when this exception occurs.

(3) XTLB Refill Exception

If an exception occurs after startup of the user boot program and BEV=0 is set to the TX49/H3 core Status Register, the TX49/H3 core transfers the control to address 0x8000_0080.

When BEV=1, NAND IPL Exception Handler refers to the Jump Address Register (JMPADR) from exception vector address 0xBFC0_0280 and transfers the control to the (JMPADR) + 0x0280 address of the user boot program. Therefore, you need to describe an exception vector to the (JMPADR) + 0x0280 address of the user boot program when this exception occurs.

(4) Other Exceptions (Common Exceptions)

If an exception occurs after startup of the user boot program and BEV=0 is set to the TX49/H3 core Status Register, the TX49/H3 core transfers the control to address 0x8000_0180.

When BEV=1, NAND IPL Exception Handler refers to the Jump Address Register (JMPADR) from exception vector address 0xBFC0_0380 and transfers the control to the (JMPADR) + 0x0380 address of the user boot program. Therefore, you have to describe an exception vector to the (JMPADR) + 0x0380 address of the user boot program when this exception occurs.

For details on interrupts, see the document "64-bit TX System RISC TX49/H3 Core Architecture" provided by Toshiba.

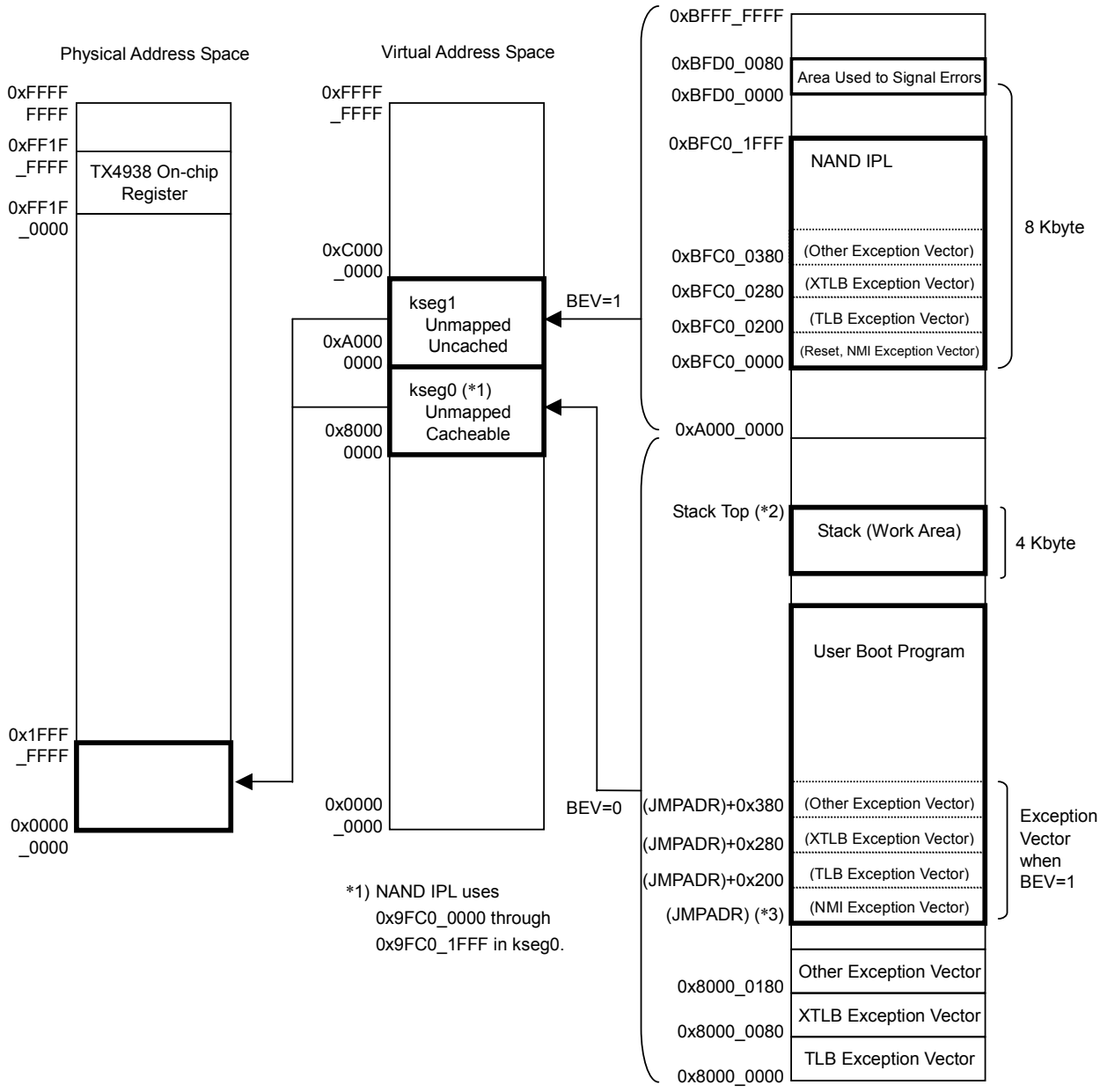


Figure 18.3.2 NAND IPL Address Map

18.3.4 NAND Flash Memory IPL Control Information

18.3.4.1 IPL Control Information Configuration

To start up NAND IPL, you need to save the IPL control information in the NAND flash memory in advance. Describe the IPL control information using multiple control information commands in compliance with the prescribed format. You need to describe it in the order given below.

(1) IPL Control Information Start ID

Write the "TX49" ASCII code as the start information.

(2) NAND Flash Memory Access Setup

Set up the strobe pulse width (SPW) of the NAND Flash Memory Strobe Pulse Width Register (NDFSPR).

Note: The NAND flash memory operates at the slowest speed value with the initial NAND IPL setting. If you want to accelerate NAND IPL, set a value appropriate for the speed of the NAND flash memory being used.

(3) RAM Access Setup

RAM is connected to either the External Bus Controller or the SDRAM Controller, which depends on the type of RAM to be used. You will perform initial setting of memory access required for RAM to be connected. The following shows examples when RAM is connected to the SDRAM Controller. For details, see subsection 9.3.3 Initialization of SDRAM.

– SDRAM Channel Control Register (SDCCRn)

– SDRAM Timing Register (SDCTR)

– SDRAM Command Register (SDCCMD)

– Refresh Counter Setup

– SDRAM Intialization

(4) Stack (Work Area) Setup

Set the address of the 4-Kbyte stack (work area) used by NAND IPL.

(5) Page Crossing Setup

"Page crossing" refers to a situation where the IPL control information can not be contained within a single page (512 bytes) and spans across multiple pages. When performing page crossing, the "page crossing command" indicates that the subsequent command resides on the next page. Write a "IPL control information continuation ID" on the top of the next page. When the IPL control information is contained in a single page, this command is not required.

(6) User Boot Program Setup

Specify the logical address and program size (bytes) of the NAND flash memory that saves the user boot program, and the load destination address of RAM.

Note: To use RAM with memory parity or ECC, initialize the user boot program area of the load destination because reading RAM having no data results in an error.

18.3.4.2 Describing the IPL Control Information

Describe the IPL control information in NAND flash memory in the following format.

- Describe it in the format of the command or the parameters that the command takes. For details, see Table 18.3.2.
- Commands and parameters are in units of 32 bits/data.
- Write the same command or parameter three times in a row. In other words, write the same 32-bit data three times in a row.
- Always describe it in the Big-endian format regardless of endian during operation.

Table 18.3.2 IPL Control Information Command

No.	Command Name	Command Value	Function	No. 1 Parameter	No. 2 Parameter	No. 3 Parameter
1	STARTID	0x5458_3439	IPL Control Information Start ID	-	-	-
2	WRITE32	0x0000_0001	32-bit Write	Address	Data	-
3	WRITE64	0x0000_0002	64-bit Write	Address	High Data	Low Data
4	WAITSDRAM	0x0000_0003	SDRAM Initialization Completion Wait	0x0000_0000	-	-
5	WAITTIME	0x0000_0004	Wait Time	Loop Count	-	-
6	MEMORY	0x0000_0005	Memory Initialization	Start Address	Size (Bytes)	-
7	WORK	0x0000_0006	Specify Stack	Stack Top	-	-
8	NEXTINF	0x0000_0007	Page Crossing	0xEEEE_EEEE	-	-
9	NEXTID	0x5458_346E	IPL Control Information Continuation ID	-	-	-
10	BOOTLA	0x0000_0008	User Boot Program Specification	NAND Flash Memory Logical Address	Program Size (Bytes)	Load Destination RAM Address
11	NOP	0x0000_0009	No Operation	0x0000_0000	-	-

(1) IPL Control Information Start ID (STARTID)

This command indicates start of the IPL control information. Describe "TX49" (0x5458_3439) in ASCII code. NAND IPL recognizes data from this command to the BOOTLA command, which is written in NAND flash memory, as the IPL control information.

(2) 32-bit Write (WRITE32)

Use this command to perform operations such as SDRAM initialization using register settings. Write 32-bit data to the specified 32-bit address.

(3) 64-bit Write (WRITE64)

Use this command to perform operations such as SDRAM initialization for register setting. Combine 32-bit high-order data and 32-bit low-order data into one 64-bit data, and write the 64-bit data to the specified 32-bit addresses.

Note: You can use the 32-bit Write command to initialize a 64-bit register, but we recommend to use the 64-bit write command.

(4) SDRAM Initialization Completion Wait (WAITSDRAM)

This command waits until the count value set in the Refresh Counter (RC) of the SDRAM Timing Register (SDCTR) becomes 0. Before invoking this command, set the refresh counter value required for initialization in accordance with the specifications of the SDRAM being used.

(5) Wait Time (WAITTIME)

NAND IPL executes the loop among three instructions the number of times specified with this command (operation with the Instruction cache On), and waits. Note that wait time changes depending on the CPU clock.

(6) Memory Initialization (MEMORY)

This command initializes memory to 0 by specifying the address and size (in bytes) of the target memory. Use this command to initialize RAM with memory parity or ECC.

(7) Stack (Work Area) Specifications (WORK)

After enabling RAM operation, this command specifies the top address of the 4-Kbyte stack used by NAND IPL. Refer to Figure 18.3.2 NAND IPL Address Map.

(8) Page Crossing (NEXTINF)

This command is used when the IPL control information spans across multiple pages (exceeds 512 bytes in size). Describing this command indicates that the IPL control information described on this page ends. To use the NEXTINF command, the IPL Control Information Continuation command (NEXTID) is required.

(9) IPL Control Information Continuation ID (NEXTID)

This command is used when the IPL control information spans across multiple pages (exceeds 512 bytes in size). Describing ASCII code “TX4n” (0x5458_346E), that is the NEXTID command, at the top of the applicable page, indicates that this page is a continuation of the IPL control information. To use the NEXTID command, the Page Crossing command (NEXTINF) is required.

(10) User Boot Program Specification (BOOTLA)

This command specifies the logical start address of the NAND flash memory containing the user boot program (see 18.3.5.1), the size of the program (in bytes), and the load destination address of RAM to be loaded by NAND IPL.

Since NAND IPL recognizes this command as the end of the IPL control information, subsequent commands are not executed.

(11) No Operation (NO)

This command performs no operation.

18.3.4.3 Example Definition of the IPL Control Information

Table 18.3.3 Page Configuration Example of the IPL Control Information

Offset Address	Command	Content	Write Data (Example)
0x000	STARTINF	Indicates “IPL Control Information Start ID”	0x5458_3439
0x004	↓	↓	↓
0x008	↓	↓	↓
0x00c	WRITE64	Indicates “64-bit Write Command”	0x0000_0002
0x010	↓	↓	↓
0x014	↓	↓	↓
0x018	No. 1 Parameter	NDFSPR Register Address	0xFF1F_5028
0x01c	↓	↓	↓
0x020	↓	↓	↓
0x024	No. 2 Parameter	Pulse Width Setting (High Data) of NAND Flash Memory	0x0000_0000
0x028	↓	↓	↓
0x02c	↓	↓	↓
0x030	No. 3 Parameter	Pulse Width Setting (Low Data) of NAND Flash Memory	0x0000_0045
0x034	↓	↓	↓
0x038	↓	↓	↓
0x03c	WRITE64	Indicates “64-bit Write Command”	0x0000_0002
0x040	↓	↓	↓
0x044	↓	↓	↓
0x048	No. 1 Parameter	SDCCR0 Register Address	0xFF1F_8000
0x04c	↓	↓	↓
0x050	↓	↓	↓
0x054	No. 2 Parameter	SDCCR0 Register Setting (High Data)	0x0000_000E
0x058	↓	↓	↓
0x05c	↓	↓	↓
0x060	No. 3 Parameter	SDCCR0 Register Setting (Low Data)	0x0000_0404
0x064	↓	↓	↓
0x068	↓	↓	↓
0x06c	WRITE64	Indicates “64-bit Write Command”	0x0000_0002
0x070	↓	↓	↓
0x074	↓	↓	↓
0x078	No. 1 Parameter	SDCTR Register Address	0xFF1F_8040
0x07c	↓	↓	↓
0x080	↓	↓	↓
0x084	No. 2 Parameter	SDCTR Register Setting (High Data)	0x0000_0003
0x088	↓	↓	↓
0x08c	↓	↓	↓
0x090	No. 3 Parameter	SDCTR Register Setting (Low Data)	0xBE02_0400
0x094	↓	↓	↓
0x098	↓	↓	↓
0x09c	WRITE64	Indicates “64-bit Write Command”	0x0000_0002
0x0a0	↓	↓	↓
0x0a4	↓	↓	↓
0x0a8	No. 1 Parameter	SDCCMD Register Address	0xFF1F_8058
0x0ac	↓	↓	↓
0x0b0	↓	↓	↓
0x0b4	No. 2 Parameter	SDCCMD Register Setting (High Data)	0x0000_0000
0x0b8	↓	↓	↓
0x0bc	↓	↓	↓
0x0c0	No. 3 Parameter	SDCCMD Register Setting (Low Data)	0x0000_0013

	Command	Content	Write Data (Example)
0x0c4	↓	↓	↓
0x0c8	↓	↓	↓
0x0cc	WRITE64	Indicates “64-bit Write Command”	0x0000_0002
0x0d0	↓	↓	↓
0x0d4	↓	↓	↓
0x0d8	No. 1 Parameter	SDCCMD Register Address	0xFF1F_8058
0x0dc	↓	↓	↓
0x0e0	↓	↓	↓
0x0e4	No. 2 Parameter	SDCCMD Register Setting (High Data)	0x0000_0000
0x0e8	↓	↓	↓
0x0ec	↓	↓	↓
0x0f0	No. 3 Parameter	SDCCMD Register Setting (Low Data)	0x0000_0011
0x0f4	↓	↓	↓
0x0f8	↓	↓	↓
0x0fc	WAITSDRAM	Indicates “SDRAM Initialization Completion Wait Command”	0x0000_0003
0x100	↓	↓	↓
0x104	↓	↓	↓
0x108	No. 1 Parameter	Counter Value 0	0x0000_0000
0x10c	↓	↓	↓
0x110	↓	↓	↓
0x114	MEMORY	Indicates “Memory Initialization Command”	0x0000_0005
0x118	↓	↓	↓
0x11c	↓	↓	↓
0x120	No. 1 Parameter	Start Address	0xA001_0000
0x124	↓	↓	↓
0x128	↓	↓	↓
0x12c	No. 2 Parameter	Size (Bytes)	0x0000_8000
0x130	↓	↓	↓
0x134	↓	↓	↓
0x138	WORK	Indicates “Stack Specification Command”	0x0000_0006
0x13c	↓	↓	↓
0x140	↓	↓	↓
0x144	No. 1 Parameter	Stack Top Address	0xA000_8000
0x148	↓	↓	↓
0x14c	↓	↓	↓
0x150	NEXTINF	Indicates “Page Crossing Command”	0x0000_0007
0x154	↓	↓	↓
0x158	↓	↓	↓
0x15c	No. 1 Parameter	Page Crossing End ID	0xEEEE_EEEE
0x160	↓	↓	↓
0x164	↓	↓	↓
	Omitted		
0x1f0	None	None	0x0000_0000
0x000	NEXTID	Indicates “IPL Control Information Continuation ID”	0x5458_346E
0x004	↓	↓	↓
0x008	↓	↓	↓
0x00c	BOOTLA	Indicates “User Boot Program Specification Command”	0x0000_0008
0x010	↓	↓	↓
0x014	↓	↓	↓
0x018	No. 1 Parameter	NAND Flash Memory Logical Address	0x0000_0001
0x01c	↓	↓	↓
0x020	↓	↓	↓
0x024	No. 2 Parameter	Size of User Boot Program (Bytes)	0x0000_8000
0x028	↓	↓	↓

Next Page

	Command	Content	Write Data (Example)
0x02c	↓	↓	↓
0x030	No. 3 Parameter	Load Destination SDRAM Address	0xA001_0000
0x034	↓	↓	↓
0x038	↓	↓	↓

18.3.5 Data configuration of NAND Flash Memory

This section describes how to use the NAND flash memory of NAND IPL. Subsection 18.3.5.1 describes the standard physical format of the NAND flash memory prescribed in SmartMedia™ Standard 2000, and 18.3.5.2 describes how to used the NAND flash memory referred by NAND IPL.

Note: NAND IPL is designed by referring the physical format of SmartMedia™ Standard 2000, but all memory area of the NAND flash memory used by NAND IPL need not satisfy this specification. The data configuration referred by NAND IPL is described 18.3.5.2 or later subsections.

18.3.5.1 Data Configuration of the NAND Flash Memory

The standard data format (physical format) of the NAND flash memory is shown in Table 18.3.4. Since the number of zones and blocks depends on the capacitance of the NAND flash memory, use Table 18.3.4 together with Table 18.3.1. Table 18.3.5 and 18.3.6 show Card Information Structure (CIS)/Identify Drive Information (IDI) area, data area and data configuration of each redundant area.

This chapter refers to “SmartMedia™ Physical Format Specification”.

Table 18.3.4 NAND Flash Memory Physical Format Specification

Zone 0	Block 0	Page 0	CIS/ IDI Area (*1)	Redundant Area (16 bytes)
		Page 1		
		:		
		Page end		
	Block 1	Page 0	Data Area (512 bytes) (*1)	Redundant Area (16 bytes)
		Page 1	↓	↓
		:	:	:
		Page end	↓	↓
	:	:	:	:
	Block end	Page 0	↓	↓
		Page 1	↓	↓
		:	:	:
		Page end	↓	↓
:	:	:	:	:
Zone end	Block 0	Page 0	↓	↓
		Page 1	↓	↓
		:	:	:
		Page end	↓	↓
	Block 1	Page 0	↓	↓
		Page 1	↓	↓
		:	:	:
		Page end	↓	↓
	:	:	:	:
	Block end	Page 0	↓	↓
		Page 1	↓	↓
		:	:	:
		Page end	↓	↓

(*1) Supplemental Remarks

In the case of the flash memory type, the CIS/IDI area is allocated to the physical block 0. If the physical block 0 is a bad one, the CIS/IDI area is allocated to the physical block 1 or the first subsequent block that is found good. Only one block is allocated to the physical block 0, and the other blocks are used as the data areas. For the relationship between logical and physical blocks, see Table 18.3.5.

When the Mask ROM type is used, the CIS/IDI area is allocated to the physical block 0. The relationship between logical and physical blocks is "(logical block address) = (physical block address) - 1". The area of the physical block 1 through the data end block address becomes the data area, and the others are the reserved areas.

Table 18.3.5 Data Specification of the CIS/IDI Area in NAND Flash Memory

		Flash Memory Type	Mask ROM Type
	Byte	Description	Description
Data Area	0 - 127	CIS Area-1	CIS Area-1
	128 - 255	IDI Area-1	IDI Area-1
	256 - 383	CIS Area-2	CIS Area-2
	384 - 511	IDI Area-2	IDI Area-2
Redundant Area	512	Functional Select Information Area	0xFF (Fixed)
	513		0xFF (Fixed)
	514		0xFF (Fixed)
	515		0xFF (Fixed)
	516	Invalid Data Flag Area (Information Validity)	0xFF (Fixed)
	517	0xFF (Fixed)	0xFF (Fixed)
	518	0x0000 (Fixed)	0xFF (Fixed)
	519		0xFF (Fixed)
	520	ECC Area-2	0xFF (Fixed)
	521		0xFF (Fixed)
	522		0xFF (Fixed)
	523	0x0000 (Fixed)	0xFF (Fixed)
	524		0xFF (Fixed)
	525	ECC Area-1	0xFF (Fixed)
	526		0xFF (Fixed)
	527		0xFF (Fixed)

(1) CIS Area (Field)

This area contains the data required to check whether the format is based on the physical format specification of SmartMedia™ Standard 2000. The first 10 bytes are fixed to 0x01_03_D9_01_FF_18_02_DF_01_20. The same data is stored in the CIS Area-1 and CIS Area-2 to ensure reliability.

Note: We recommend to store the data in the CIS area, though NAND IPL does not use the data in it.

(2) IDI Area (Field)

This is the area to store the data used with the system having ATA interface. The same data, of which default value is 0x00, is stored in the CIS Area-1 and CIS Area-2 to ensure reliability.

Note: NAND IPL does not use the data in the IDI area.

(3) Data Stored in the Redundant Area

General information on the redundant area is shown below.

- Functional Select Information Area: The area reserved for function expansion
- Invalid Data Flag Area: The area which indicates the validity of the data in the CIS/IDI area
- ECC Area-1, ECC Area-2: 3-byte ECC code of the page data (256 bytes)

Note: NAND IPL does not use the data stored in the redundant area of the CIS/IDI area.

Table 18.3.6 Data Specification of the Data Area in NAND Flash Memory

	Flash Memory Type		Mask ROM Type
	Byte	Description	Description
Data Area	0 - 255	Data Area-1	Data Area-1
	256 - 511	Data Area-2	Data Area-2
Redundant Area	512	Reserved Area	0xFF (Fixed)
	513		0xFF (Fixed)
	514		0xFF (Fixed)
	515		0xFF (Fixed)
	516	Data Status Area	0xFF (Fixed)
	517	Block Status Area	0xFF (Fixed)
	518	Block Address Area-1	0xFF (Fixed)
	519		0xFF (Fixed)
	520	ECC Area-2	0xFF (Fixed)
	521		0xFF (Fixed)
	522		0xFF (Fixed)
	523	Block Address Area-2	0xFF (Fixed)
	524		0xFF (Fixed)
	525	ECC Area-1	0xFF (Fixed)
	526		0xFF (Fixed)
	527		0xFF (Fixed)

Note: For the data in the redundant area of the data area used by NAND IPL, see Table 18.3.10.

(1) Data Area (Field)-1

Contains the data of 0 to 255th bytes, that is the first half of the 512-byte data.

(2) Data Area (Field)-2

Contains the data of 256th to 512th bytes, that is the last half of the 512-byte data.

(3) Reserved Area (Field)

This is the area reserved for function expansion, and contains the fixed value 0xFFFF_FFFF.

(4) Data Status Area (Field)

This area indicates that Data Area-1 and Data Area-2 store incorrect data. 0xFF is set typically. When incorrect data is written, 0x00 is set to this area.

(5) Block Status Area (Field)

This area indicates whether the block is good or not. 0xFF (good block) is set typically. In the case of the bad block, 0x00 (initial bad block) or 0xF0 (subsequent bad block) is set to this area. Specify the same value in the same block.

(6) Block Address Area (Field)-1

This area indicates translation information from physical block address into the logical block address. Table 18.3.7 shows the data specification. Specify the same value in the same block. For details on the logical-physical address translation, see "logical-physical address translation table" described later.

Table 18.3.7 Data Specification in Block Address Area

D7	D6	D5	D4	D3	D2	D1	D0	512 + 16 Byte/Page
0	0	0	1	0	BA9	BA8	BA7	518,523 Byte
BA6	BA5	BA4	BA3	BA2	BA1	BA0	P	519,524 Byte

BA9 ~ BAO: Block Address (Value = 0 ~ 'Max Logical Blocks 1') P: Even-parity Bit

Note: Block Address indicates the logical address partitioned in erase block units, that is 8 bytes.

(7) Block Address Area (Field)-2

This area contains the data same as Block Address Area-1.

<Logical-Physical Address translation Table>

The SmartMedia™ Standard 2000 physical format specification defines that the logical block in NAND flash memory is not assigned to a specific physical block. The Block Address Area in each block contains logical block addresses assigned to physical blocks. Logical-physical address translation is available by searching these addresses. A physical and logical blocks are paired. Since all logical blocks are not assigned to physical blocks, logical blocks reside to which physical blocks are not assigned. Figure 18.3.3 shows an example of a logical-physical address translation table.

This subsection refers to "SmartMedia™ Interface library (SMIL) -Software Version-".

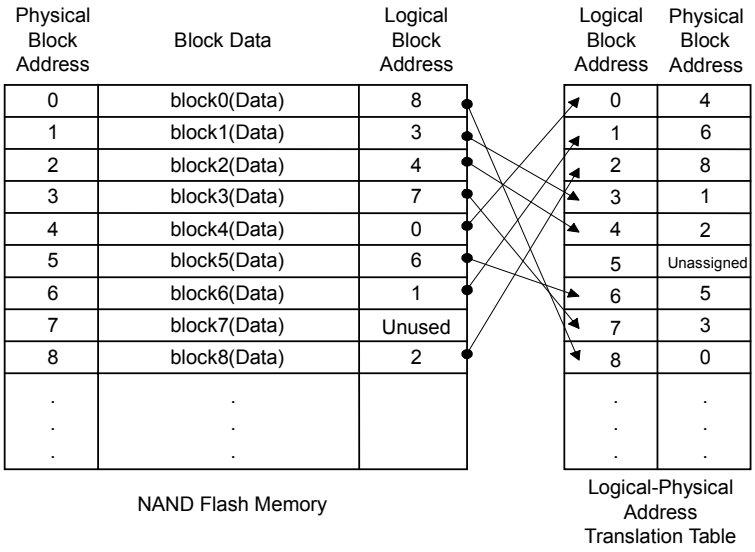


Figure 18.3.3 Logical-Physical Address Translation Table

<Supplemental Remarks for Logical Addresses>

- The number of physical blocks in one zone is 1024, and that of logical blocks is 1000.
- The value described in the Block Address Area is 0 to 999.
- The logical block addresses can not be assigned to the physical block containing the CIS/IDI area.
- The logical block address is a unique address, which is the sequential number from the logical block 0 in an entire NAND flash memory. The address is obtained with the following equation.

  Logical block address = (*Zone_number* × 1000)
  + (*Block_number_described_in_Block_Address_Area*)

- The logical page address is the sequential page address which starts from the page 0 in the logic block 0 and ends with the last page in the last logic block. The address is obtained with the following equation.

  Logical page address = (*Logical_block_address* × *the_number_of_pages_in_each_block*)
  + (*Page_number_described_in_the_block*)

(8) ECC AREA (Field)-1

This area presents the 3-byte ECC code for the page data which is assigned to 0 through 255th byte (256 bytes). For details on ECC code, see “ECC Method” described below.

(9) ECC AREA (Field)-2

This area presents the 3-byte ECC code of the page data which is assigned to 256th through 511th byte (256 bytes). For details on ECC code, see “ECC Method” described below.

<ECC Methodology>

256-byte data area is added with 22-bit ECC code by the physical format specification of SmartMedia™ Standard 2000. Data Area-1 and Data Area-2 are added with ECC Area-1 and ECC Area-2, respectively. The EEC code provides correction capability of 1-bit random error, and detection capability of 2-bit random error. EEC code is generated for the data area, and not for the data in the page redundant area including ECC code. The data in the page redundant area is duplicate to ensure reliability, which enables you to check the correctness of the data. 256-byte data is handled as 2048-bit serial data by ECC equation. When an error occurs, the error occurrence bit is detected with the parity data information, and data can be corrected.

- Defining Address of 2048-bit Serial Data

  256 bytes are aligned to 2048-bit serial data as shown in Table 18.3.8.

Table 18.3.8 2048-bit Serial Data Address

	bit 7	bit 6					bit 1	bit 0
1st Byte	00000000 111	00000000 110					00000000 001	00000000 000
2nd Byte	00000001 111	00000001 110					00000001 001	00000001 000
255th Byte	11111110 111	11111110 110					11111110 001	11111110 000
256th Byte	11111111 111	11111111 110					11111111 001	11111111 000

The top-line in the above table indicates input of the first byte, and the bottom-line is input of the 256th byte. The bit 0 of the first bite becomes the first bit of 2048 bits (address 00000000 000), and the bit 7 of 256th byte becomes the 2048th bit of 2048 bits (address 11111111 111).

- Generating Parity Data

  Parity data consists of 22 bits; CP0, CP1, CP2, CP3, CP4, CP5, LP00, LP01, LP02, LP03, LP04 through LP14 and LP15. The Column parity (CP) consists of 6 bits and Line Parity (LP) consists of 16 bits. The parity data to be generated is aligned as shown below and stored in the page redundant area data.

Table 18.3.9 ECC Data Alignment

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	512 + 16 Byte/Page
LP07	LP06	LP05	LP04	LP03	LP02	LP01	LP00	520,525 Byte
LP15	LP14	LP13	LP12	LP11	LP10	LP09	LP08	521,526 Byte
CP5	CP4	CP3	CP2	CP1	CP0	“1”	“1”	522,527 Byte

Line Parity (LP) and Column Parity (CP) are odd-parity of 1024 bits which satisfy the following conditions.

LP00 = D (*******0 , ***)
LP01 = D (*******1 , ***)
LP02 = D (******0* , ***)
LP03 = D (******1* , ***)
LP04 = D (*****0** , ***)
LP05 = D (*****1** , ***)
LP06 = D (****0*** , ***)
LP07 = D (****1*** , ***)
LP08 = D (***0**** , ***)
LP09 = D (***1**** , ***)
LP10 = D (**0***** , ***)
LP11 = D (**1***** , ***)
LP12 = D (*0****** , ***)
LP13 = D (*1****** , ***)
LP14 = D (0******* , ***)
LP15 = D (1******* , ***)

CP0 = D (******** , **0)
CP1 = D (******** , **1)
CP2 = D (******** , *0*)
CP3 = D (******** , *1*)
CP4 = D (******** , 0**)
CP5 = D (******** , 1**)

where, an asterisk (*) indicates 0 or 1.

18.3.5.2 Writing IPL Control Information

- Write IPL control information in the block containing the CIS area, which is assigned to the physical address 0 through 23rd block, or subsequent block. We recommend to write it in the block containing the CIS area, and in the page next to the page containing the CIS area.When IPL control information is not written since there is not good one until the physical address 23, the NAND flash memory can not be used because NAND IPL does not operate correctly.

- The redundant block data containing the IPL control information must satisfy the following conditions. See Table 18.3.10.
  - Block Status Area is correct. (0 is not set or set to 1 bit)
  - The same data is stored in Block Address Area-1 and Block Address Area-2.

18.3.5.3 Writing the User Boot Program

- The user boot program can be written in any logical address of the NAND flash memory, and span across multiple blocks and zones.

- The user boot program is handled as a single object assigned to continuous space with the logical address in the NAND flash memory. NAND IPL can load only a single object into the continuous space in RAM.

- The data in the redundant area containing the user boot program must satisfy the following conditions. See Table 18.3.10.

  Note: NAND IPL neither registers the block which does not satisfy the following conditions in the logical-physical address translation table, nor issues an error of this block because it is not loaded into RAM. See Table 18.3.10.

  – Block Status Area and Data Status Area are good ones (0 is not set or set to 1 bit)
  – The same data is stored in Block Address Area-1 and Block Address Area-2.
  – The value in Block Address Area-1 does not match to the value 0x000 in Block Address Field-s fro CIS/IDI.
  – The value in Block Address Area-1 is not a valid logical address (high-order 4 bits do not match 0001B.)
  – The even-parity bit in Block Address Area-1 is correct.

- When the same address appears multiple times in the logic block address which indicates the user boot program, NAND IPL is not guaranteed to behave correctly.

18.3.5.4 IPL Control Information and Areas Except the User Boot Program

- Any data description is allowed in the zone which is not used with NAND IPL, that is the area in which the IPL control information and the user boot program are not stored.

Note: NAND IPL creates a logical-physical address translation table based on the data in the redundant area upon accessing to NAND flash memory. The redundant area of the zone containing the IPL control information and user boot program must be written in compliant with the specification shown in 18.3.5.1.

Table 18.3.10 Data in the Redundant Area Used with NAND IPL

Redundant Block	Area	Searching IPL Control Information (*1)	Creating the Logical-Physical Translation Table (*2)	Loading the User Boot Program into RAM (*3)
512	Reserved Area	–	–	–
513				
514				
515				
516	Data Status Area	–	Used	–
517	Block Status Area	Used	Used	–
518	Block Address Area-1	Used	Used	–
519				
520	ECC Area-2	–	–	Used
521				
522				
523	Block Address Area-2	Used	Used	–
524				
525	ECC Area-1	–	–	Used
526				
527				

(*1) Used to search the IPL control information written in the NAND flash memory at the stage “7) Is IPL Control Information processed correctly?” in Figure 18.3.1.

(*2) Used to create the translation table at the stage “9) Create NAND Flash Memory Logical-Physical address Translation Table.” in Figure 18.3.1.

(*3) Used to load the user boot program into RAM at the stage “11) Load the User Boot Program.” in Figure 18.3.1.

18.3.6 NAND Flash Memory IPL Error Handling

NAND IPL signals the program operation to an external device using 1 bit of PIO[0]. PIO[0] becomes Low upon startup of NAND IPL, and High to transfer the control to the user boot program after it is loaded into RAM correctly. If PIO[0] remains Low, abnormal condition occurs.

When NAND IPL detects the abnormal condition, it sets the error ID shown in Table 18.3.4 to $4($a0) in the general register, and continues issuing the EROR ID to the low-order bit of virtual address 0xBFD0_0000 by setting the external bus controller channel 0 of 32-bit bus width (EBCCR0=0x01FD_0000_0013_E008). To signal this error, the area 0xBFD_0000 through 0xBFD0_0080 is used. This information is recognized by connection to the logic analyzer.

Even in the case the error is detected, the user boot program is invoked in the following cases.

- When detecting a 1-bit error in the ECC code, NAND-IPL corrects the 1-bit error and loads the user boot program into RAM.
- When the size of the user boot program is smaller than that written in the IPL control information, the same amount of data as that being written (user boot program + &) is loaded into RAM. PIO[0] remains Low to signal the error condition.

When the size of the user boot program is larger than that written in the IPL control information, NAND IPL does not operate correctly since it loads the same amount of data as that being written in the IPL control information.

Table 18.3.11 Errors (Address Bit)

Error ID	Address [7:0]	Error Type	Cause of Error
0x010	0x04	NAND flash memory ID error	Undefined NAND flash memory is used.
0x020	0x08	IPL control information error	The IPL control information is incorrect.
0x040	0x10	ECC 2- or more-bit error	The NAND flash memory data is corrupted. (ECC code indicates 2- or more-bit error.)
0x080	0x20	ECC 1-bit Error	The NAND flash memory data is corrupted. (ECC code indicates 1-bit error.)
0x100	0x40	User boot program error	The user boot program is not written in NAND flash memory.
0x200	0x80	NAND IPL checksum error	NAND IPL is not written to on-chip ROM correctly.

A reset or NMI exception which occurs during NAND IPL operation results in abnormal termination since the jump address register (JMPADR) addresses NAND IPL (0xBFC0_0000). When the error detailed information is not generated, NMI exception may occur.

<Considerations Required for System Design>

When PIO[0] is used for the purpose other than a signal which indicates NAND IPL operation on the system, NAND IPL does not operate as intended. That must be avoided on the system.

PIO[0] is multiplexed with SPICLK on TX4938. To use SPICLK, you need to establish connection of PIO[0] so that no electrical problem occurs when NAND IPL outputs PIO[0] upon startup of system.

19. On-Chip SRAM

19.1 Characteristics

The TX4938 has 1-channel of on-chip, fast 2-KB SRAM. You can map it to any address space by setting the Base Address Register. You can access SRAM from each G-Bus Master: the TX49/H3 core, DMAC0, DMAC1, PCI0, and PCI1.

(1) 2 KB × 1 channel
(2) Specifiable base address
(3) Accessible in Byte, half-word, word, or double-word units
(4) Supports Burst access[1]
(5) Supports the critical word first function of the TX49/H3 core

19.2 Block diagram

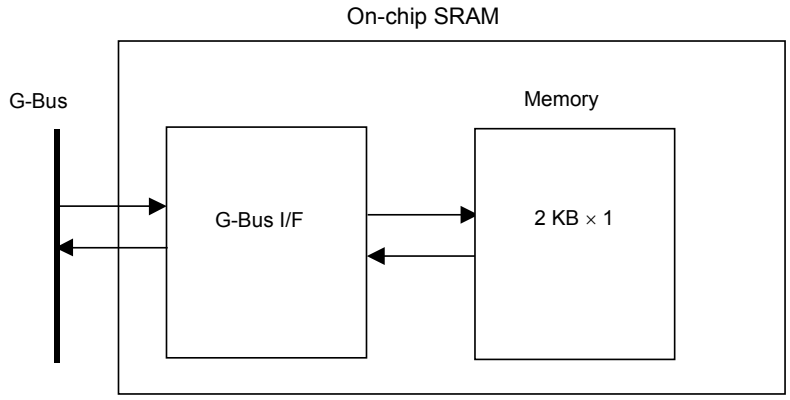


Figure 19.2.1 On-chip SRAM Block Diagram

[1] On-chip SRAM does not support special Burst access (address decrementing/fixed) of DMAC0, DMAC1.

19.3 Detailed explanation

19.3.1 Base address

Specify the base address of the on-chip SRAM in 2-KB boundary units (specify the upper 25 bits of the physical address). Operation is not guaranteed if the address overlaps with an address space set by SDRAMC, EBUSC, etc.

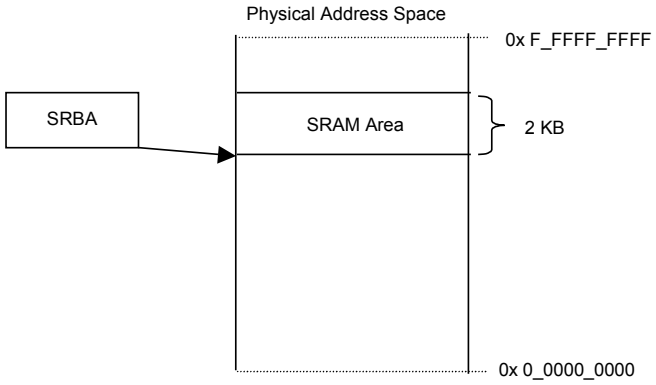


Figure 19.3.1 Base Address Specification

19.3.2 Access cycle count

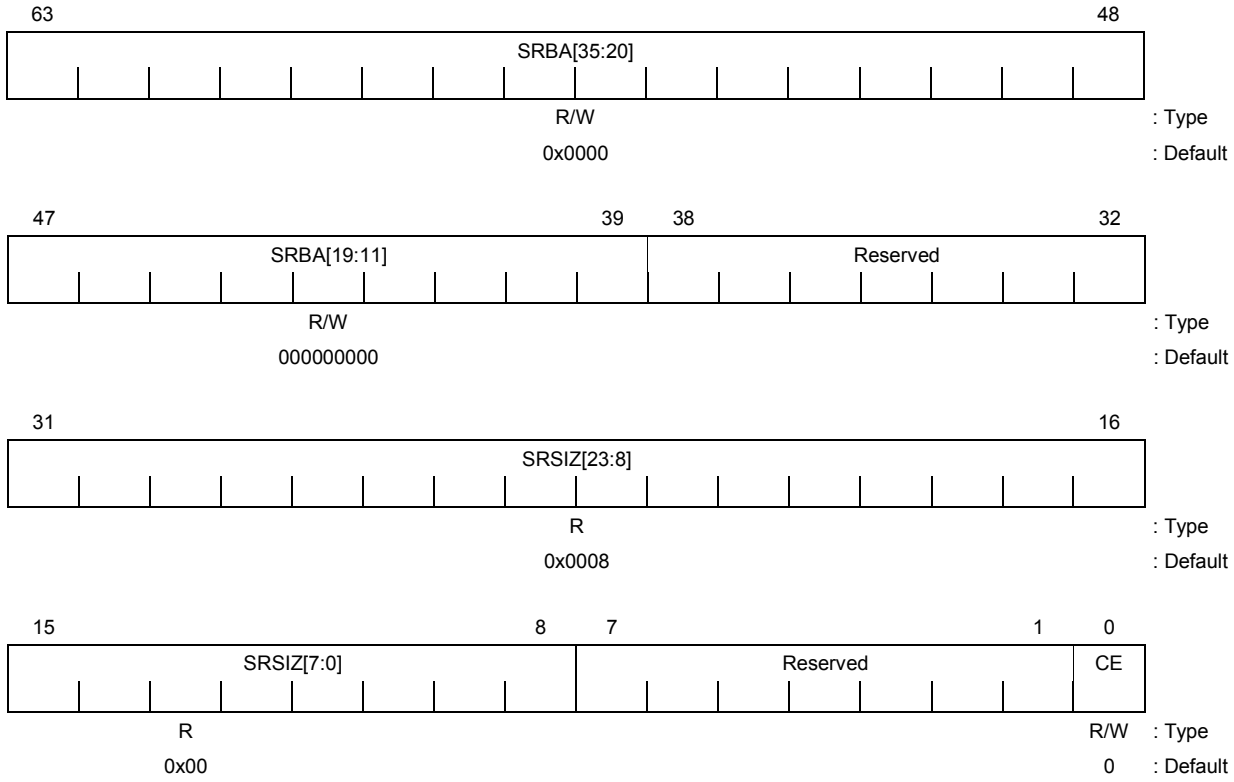
The following table shows the GBUSCLK cycle count required to access on-chip SRAM.

Table 19.3.1 Access Cycle Count

Access Type	GBUSCLK Cycle Count (n= Burst Size (1-8))
Single Read	3
Single Write	3
Burst Read	2+n
Burst Write	2+n

19.4 Register

19.4.1 On-chip SRAM Control Register (SRAMCR : 0x6000)



Bit(s)	Mnemonic	Field Name	Description
63:39	SRBA [35:11]	On-chip SRAM Base Address	SRAM Base Address (Default: 0, R/W) This field specifies the base address of on-chip SRAM. The upper 25 bits [35:11] of the physical address are compared with the value of this field. A 2-KB physical address space with the base address SRBA0[35:11] is mapped to on-chip SRAM.
38:32		Reserved	
31:8	SRSIZ [23:0]	On-chip SRAM Size	SRAM Size (Default: 0x00_0800, R) Displays the on-chip SRAM size (2 Bytes).
7:1		Reserved	
0	CE	Channel Enable	Channel enable (0 , R/W) This bit specifies whether to enable a channel. When using on-chip SRAM, set "1". 0: Disable 1: Enable

Figure 19.4.1 On-Chip SRAM Control Register

20. NAND Flash Memory Controller

20.1 Features

The TX4938 has a built-in NAND Flash Controller (NDFMC). The NDFMC generates the control signals required to interface with NAND flash memory. Also, the NDFMC performs ECC calculation.

The NAND Flash Controller has the following features.

- Indirect register access to NAND flash memory
- On-chip ECC calculation circuit

20.2 Block diagram

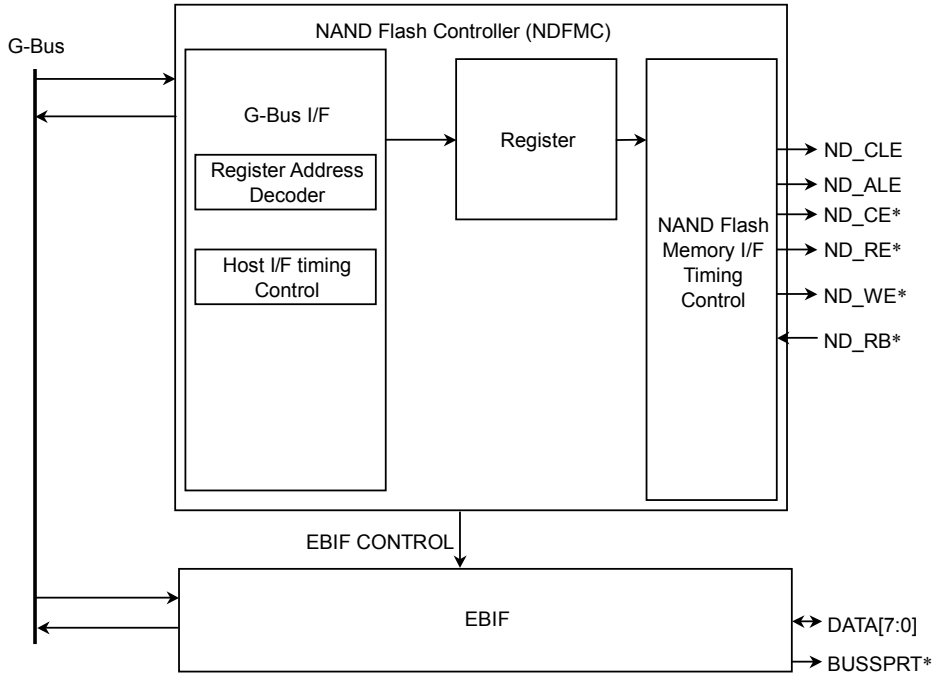


Figure 20.2.1 NAND Flash Memory Controller Block Diagram

20.3 Detailed operation

20.3.1 Accessing NAND flash memory

With the NDFMC, you can access NAND flash memory by controlling registers. The NDFMC also has an ECC calculation function. See 20.3.2 for more information on ECC. This subsection describes the procedure for accessing NAND flash memory.

Basically, first set a command in the NDFMCR, then either read or write NDFDTR to access NAND flash memory. Reading NDFDTR executes a read cycle to external NAND flash memory. The read cycle to NDFDTR also ends when the read cycle to external NAND flash memory ends. Similarly, writing NDFDTR executes a write cycle to external NAND flash memory. The write cycle to NDFDTR also ends when the write cycle to external NAND flash memory ends.

Note: When reading or writing NDFDTR, the ND_RE* or ND_WE* signal is asserted. The low pulse width and high-level time after the ND_RE* or ND_WE* signal is deasserted can be set in NDFSPR. Other signals are controlled by software to meet AC specifications of NAND flash memory to be used. Be aware of the following AC specifications.

tWB: Time from ND_WE* High to assertion of ND_RB* Low. (After writing NDFDTR, wait tWB time before reading NDFSR.BUSY, then detect timing of ND_RB* assertion and deassertion. When NDFIMR.MRDY = 1 is set, NDFISR.RDY is set to 1 if ND_RB* changes from Low to High. You can detect deassertion of ND_RB* by checking this register.)

tAR1: Time from ND_ALE Low to ND_RE* Low (ID read)
(After writing NDFMCR.ALE to 0, wait tAR1 time and read NDFDTR.)

tAR2: Time from ND_ALE Low to ND_RE* Low (Read cycle)
(After writing NDFMCR.ALE to 0, wait tAR2 time and read NDFDTR.)

20.3.1.1 Initialization

Following is the initialization sequence.

1. NDFSPR (0x5028): Set the low pulse width.
2. NDFIMR (0x5020): Set 0x81 to enable interrupts.

20.3.1.2 Write

Following is the write sequence.

1. NDFMCR(0x5008): Set to 0x70 and initialize the ECC data of NDFMC.
2. The NDFMC does not have a WP* signal. Use either PIO or an external circuit to set the WP* signal to "H".
3. 512-Byte write
   - NDFMCR (0x5008): Set to 0x91, assert the ND_CLE* signal, then enter the Command mode.
   - NDFDTR (0x5000): Write 0x80 (Serial Data Input command).
   - NDFMCR (0x5008): Set to 0x92, assert the ND_ALE* signal, then enter the Address mode.
   - NDFDTR (0x5000): Set A[7:0], A[16:9], A[24:17] (and A[25] if necessary) in order.
   - NDFMCR (0x5008): Set to 0xb0 and enter the Data mode.
   - NDFDTR (0x5000): Set 512-Byte data in order.
4. Read ECC data
   - NDFMCR (0x5008): Set to 0xd0, and enter the ECC Data Read mode.
   - NDFDTR (0x5000): Read the 6-Byte ECC data the NDFMC calculated.

     First data: LPR[7:0]

     Second data: LPR[15:8]

     Third data: CPR[5:0], 2'b11

     Fourth data: LPR[23:16]

     Fifth data: LPR[31:24]

     Sixth data: CPR[11:6], 2'b11
5. Write 16-Byte redundant data
   - NDFMCR (0x5008): Set to 0x90 and enter a Data mode that does not calculate ECC.
   - NDFDTR (0x5000): Write 16-Byte redundant data.

     D520: LPR[23:16]

     D521: LPR[31:24]

     D522: CPR[11:6], 2'b11

     D525: LPR[7:0]

     D526: LPR[15:8]

     D527: CPR[5:0], 2'b11
6. Execute a page program
   - NDFMCR (0x5008): Set 0x91, assert the ND_CLE* signal, and enter the Command mode.
   - NDFDTR (0x5000): Write 0x10 (Page Program command).
   - NDFMCR (0x5008): Set 0x10 and deassert the ND_CLE* signal.
   - NDFSR (0x5010): Check the BUSY flag. Proceed to the next step if "0". Wait until the flag becomes "0" if it is "1".

7. Read status
   - NDFMCR (0x5008): Set 0x11, assert the ND_CLE* signal, and enter the Command mode.
   - NDFDTR (0x5000): Write 0x70 (Read Status command).
   - NDFMCR (0x5008): Set 0x10 and deassert the ND_CLE* signal.
   - NDFDTR (0x5000): Read the status data from NAND flash memory.
8. Repeat steps 1-7 on the other pages.

20.3.1.3 Read

Following is the read sequence.

1. NDFMCR (0x5008): Set 0x70 and initialize the NDFMC ECC data.
2. Read 512 bytes
   - NDFMCR (0x5008): Set 0x11, assert the ND_CLE* signal, and enter the Command mode.
   - NDFDTR (0x5000): Write 0x00 (Read command).
   - NDFMCR (0x5008): Set 0x12, assert the ND_ALE* signal, then enter the Address mode.
   - NDFDTR (0x5000): Set A[7:0], A[16:9], A[24:17] (and if necessary A[25]) in order.
   - NDFMCR (0x5008): Set 0x10, and deassert the ND_ALE* signal.
   - NDFSR (0x5010): Check the BUSY flag. Proceed to the next step if it is "0". Wait until it becomes "0" if it is "1".
   - NDFMCR (0x5008): Set 0x30, and enter a Data mode that performs ECC calculation.
   - NDFDTR (0x5000): Read 512-Byte data.
   - NDFMCR (0x5008): Set 0x10, and enter a Data mode that does not perform ECC calculation.
   - NDFDTR (0x5000): Read 16-Byte redundant data.
3. Read ECC data
   - NDFMCR (0x5008): Set 0x50, and enter the ECC Data Read mode.
   - NDFDTR (0x5000): Read the 6-Byte ECC data the NDFMC calculated.

     First data: LPR[7:0]
     Second data: LPR[15:8]
     Third data: CPR[5:0], 2'b11
     Fourth data: LPR[23:16]
     Fifth data: LPR[31:24]
     Sixth data: CPR[11:6], 2'b11
4. Compare the ECC data. If the data do not match, invoke the necessary error process.
5. Read another page.
   - NDFMCR (0x5008): Set 0x10.
   - NDFSR (0x5010): Check the BUSY flag. Proceed to the next step if "0". Wait until it becomes "0" if it is "1".
   - Repeat steps 1-4. However, when performing a sequential read, skip 2-1 through 2-5.

20.3.1.4 Read ID

Following is the ID read sequence.

1. NDFMCR (0x5008): Set 0x11, assert the ND_CLE* signal, and enter the Command mode.
2. NDFDTR (0x5000): Write 0x90 (ID Read command).
3. NDFMCR (0x5008): Set 0x12, assert the ND_ALE* signal, and enter the Address mode.
4. NDFDTR (0x5000): Write 0x00.
5. NDFMCR (0x5008): Set 0x10, and enter a Data mode that does not calculate ECC.
6. NDFDTR (0x5000): Read the maker code.
7. NDFDTR (0x5000): Read the device code.

20.3.2 ECC control

The NDFMC only performs ECC calculation. Use a program to check ECC for errors and correct any errors that occur.

You can read from the NDFDTR Register the ECC data calculated by NDFMC by setting the NDFMCR to either 0xD0 (Write mode) or 0x50 (Read mode). The ECC data you can read is 6-Byte data. Read NDFDTR six times. Data is read in the following order.

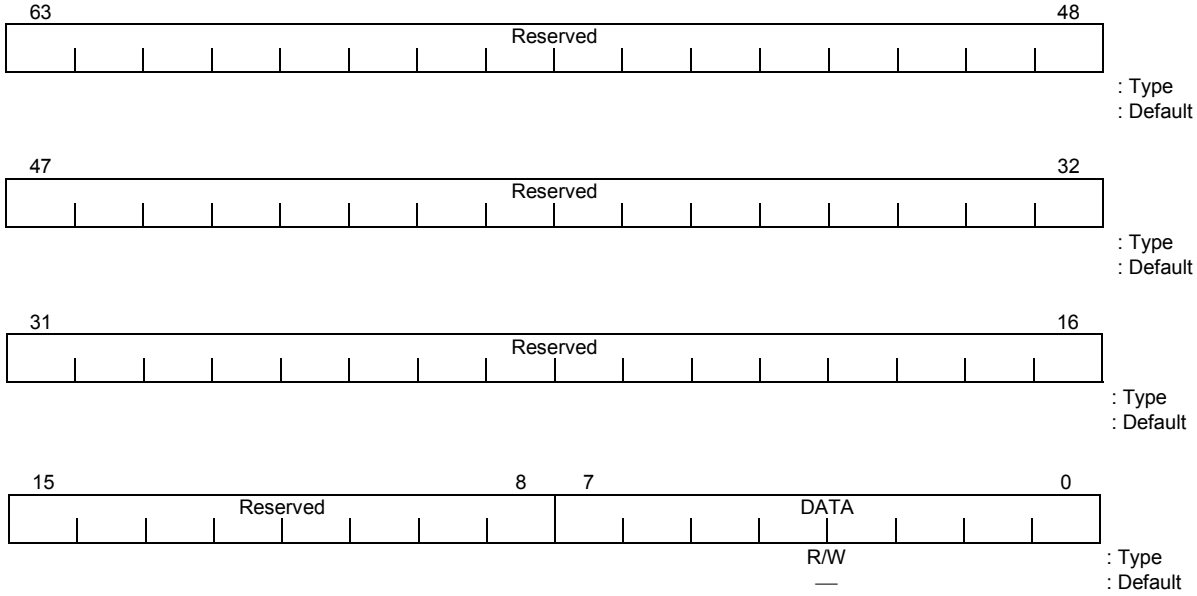
First data: LPR[7:0]
Second data: LPR[15:8]
Third data: CPR[5:0], 2'b11
Fourth data: LPR[23:16]
Fifth data: LPR[31:24]
Sixth data: CPR[11:6], 2'b11

20.4 Registers

Table 20.4.1 NDFMC Registers

Offset Address	Bit Width	Register Symbol	Register Name
0x5000	64	NDFDTR	NAND Flash Memory Data Transfer Register
0x5008	64	NDFMCR	NAND Flash Memory Mode Control Register
0x5010	64	NDFSR	NAND Flash Memory Status Register
0x5018	64	NDFISR	NAND Flash Memory Interrupt Status Register
0x5020	64	NDFIMR	NAND Flash Memory Interrupt Mask Register
0x5028	64	NDFSPR	NAND Flash Memory Strobe Pulse Width Register
0x5030	64	NDFRSTR	NAND Flash Memory Reset Register

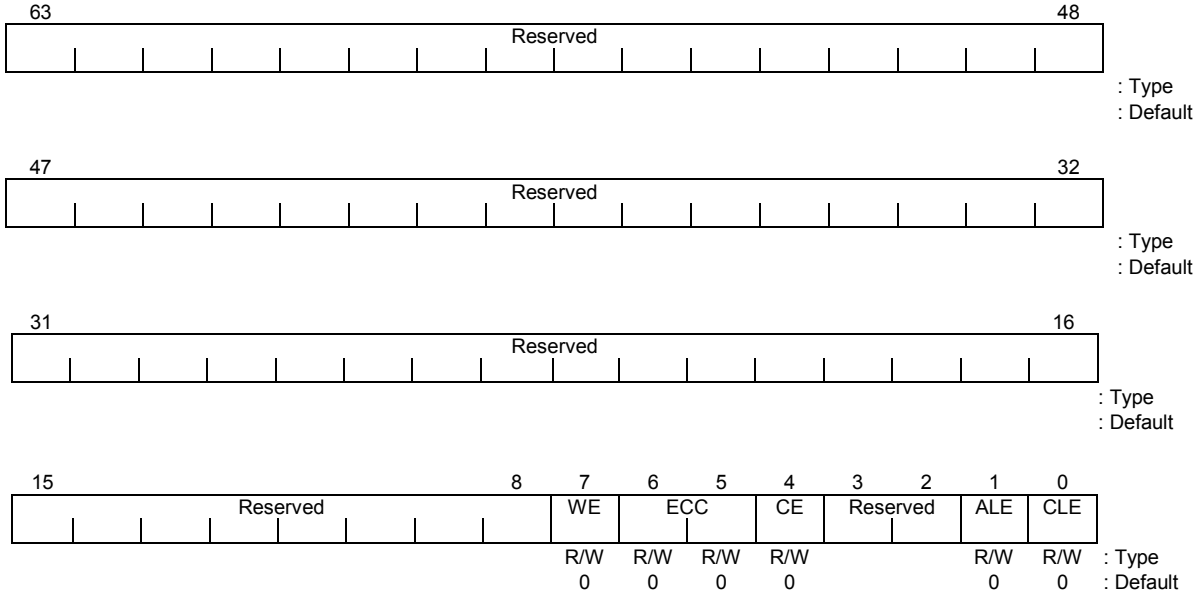
20.4.1 NAND Flash Memory Data Transfer Register (NDFDTR) 0x5000



Bits	Mnemonic	Field Name	Description
63:8	—	Reserved	—
7:0	DATA	DATA	Interrupt Detection Enable (Default: Undefined) NAND Flash memory data Read: Reads NAND flash data when in the Data mode. Write: Writes transmission data in NAND flash memory in each mode.

Figure 20.4.1 NAND Flash Memory Data Transfer Register (NDFDTR)

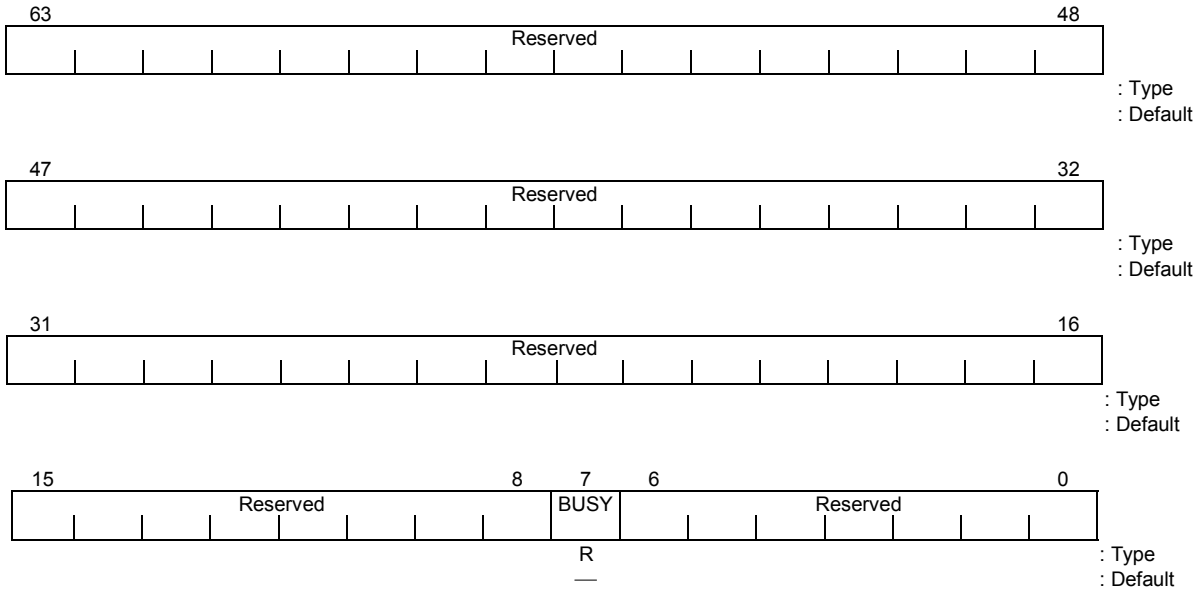
20.4.2 NAND Flash Memory Mode Control Register (NDFMCR) 0x5008



Bit(s)	Mnemonic	Field Name	Description
63:8	—	Reserved	—
7	WE	Write Enable	Write Enable (Default: 0) Enables data write operation. Set this bit when writing data to NAND flash memory. 0: Prohibit Write operation. 1: Enable Write operation.
6:5	ECC	ECC Control	ECC Control (Default: 00) This field specifies the operation of the ECC calculation circuit. 11: Reset the ECC circuit. 00: Disable the ECC circuit. 01: Enable the ECC circuit. 10: Read the ECC circuit the NDFMC calculated.
4	CE	Chip Enable	Chip Enable Default: 0) Enables access to NAND flash memory. Set this bit when accessing NAND flash memory. 0: Disable (ND_CE* high.) 1: Enable (ND_CE* low.)
3:2	—	Reserved	—
1	ALE	Address Latch Enable	Address Latch Enable (Default: 0) Specifies the ND_ALE* signal. 0: Low 1: High
0	CLE	Command Latch Enable	Command Latch Enable (Default: 0) Specifies the value of the NC_CLE signal. 0: Low 1: High

Figure 20.4.2 NAND Flash Memory Mode Control Register (NDFMCR)

20.4.3 NAND Flash Memory Status Register (NDFSR) 0x5010



Bit(s)	Mnemonic	Field Name	Description
63:8	—	Reserved	—
7	BUSY	BUSY	BUSY (Default: Undefined) Indicates the status of NAND flash memory. 0: Ready 1: Busy
6:0	—	Reserved	—

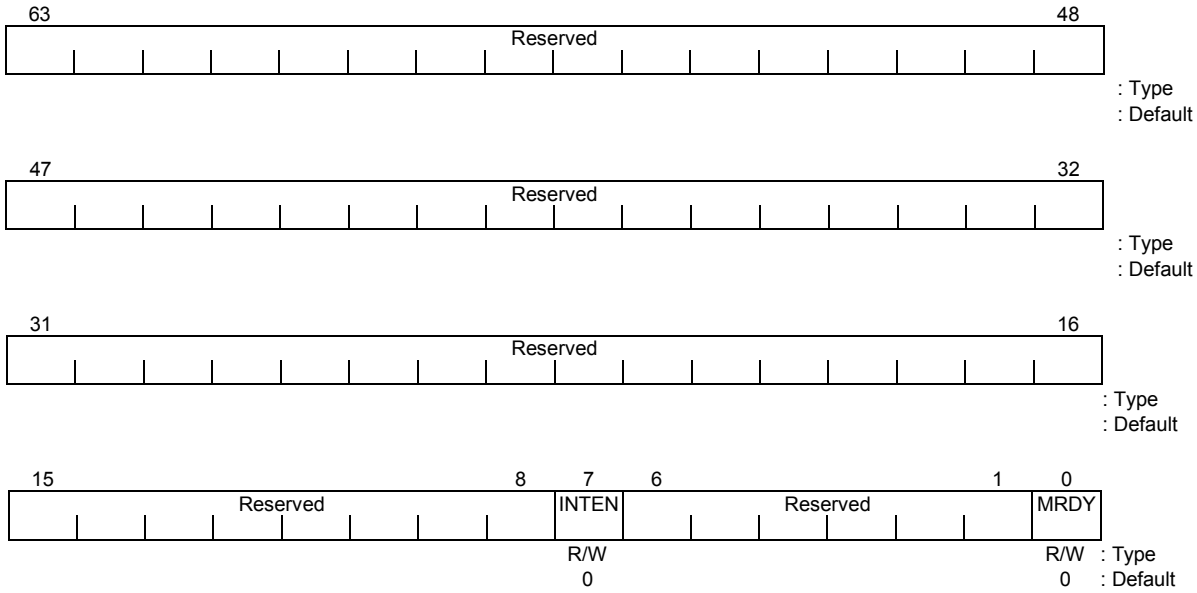
Figure 20.4.3 NAND Flash Memory Status Register (NDFSR)

20.4.4 NAND Flash Memory Interrupt Status Register (NDFISR) 0x5018

Bit(s)	Mnemonic	Field Name	Description
63:1	—	Reserved	—
0	RDY	Ready	Ready (Default: 0) When the MRDY bit of the NDFIMR Register is "1", this bit is set to "1" when the ND_RB* signal changes from Low to High. Write "1" to clear this bit to "0". Read: 0: No change 1: ND_RB* signal changed from the Busy state to the Ready state. Write: 0: Don't care 1: Clear to "0".

Figure 20.4.4 NAND Flash Memory Interrupt Status Register (NDFISR)

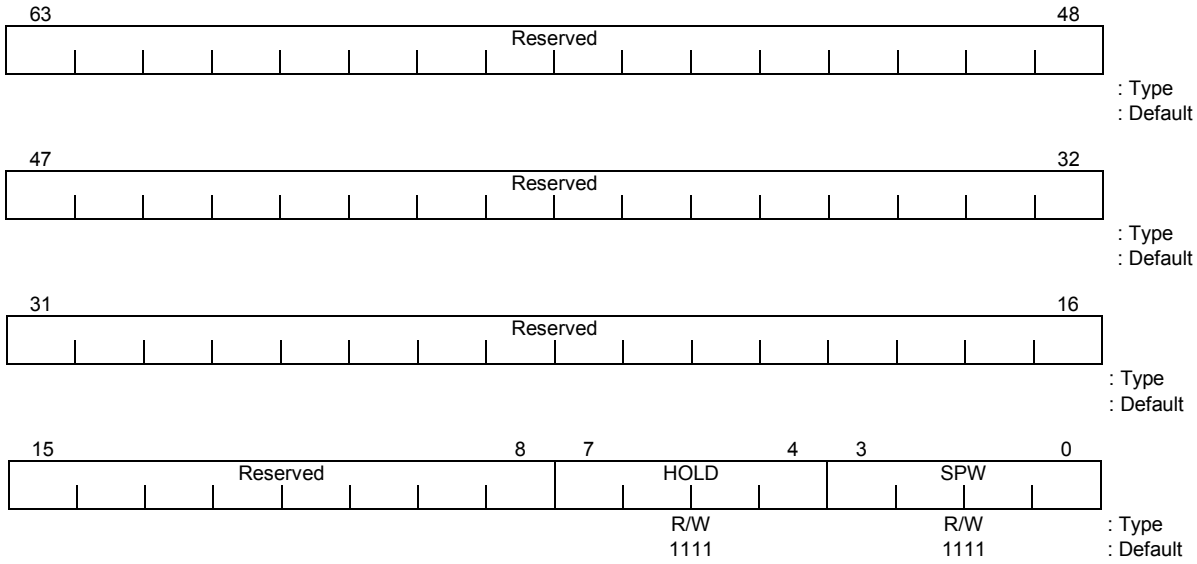
20.4.5 NAND Flash Memory Interrupt Mask Register (NDFIMR) 0x5020



Bit(s)	Mnemonic	Field Name	Description
63:8	—	Reserved	—
7	INTEN	Interrupt Enable	Interrupt Enable (Default: 0) This bit enables interrupts. If both the MRDY bit of this register and this INTEN bit are set to "1", an interrupt occurs when the RDY bit of the NDFISR Register becomes "1". 0: Disable 1: Enable
6:1	—	Reserved	—
0	MRDY	Mask RDY interrupt	Mask Ready Interrupt (Default: 0) This bit masks the RDY bit of the NDFISR Register. If this bit is "1", the RDY bit of the NDFISR Register is set when the ND_RB* signal changes from "0" to "1". 0: Disable the NDFISR RDY bit. 1: Enable the NDFISR RDY bit.

Figure 20.4.5 NAND Flash Memory Interrupt Mask Register (NDFIMR)

20.4.6 NAND Flash Memory Strobe Pulse Width Register (NDFSPR) 0x5028



Bits	Mnemonic	Field Name	Description
63:8	—	Reserved	—
7:4	HOLD	Hold Time	Hold Time (Default: 1111) The DATA[7:0], ND_CLE, ND_ALE and ND_CE* outputs of the NAND Flash Controller are held constant for HOLDxGBUSCLK cycles from the deassertion of the ND_WE* signal. Note: This field affects the read timing. The BUSSPRT* signal is deasserted after HOLDxGBUSCLK cycles from the deassertion of the ND_RE* signal. 0000: Reserved 0001: 1xGBUSCLK 0010: 2xGBUSCLK 0011: 3xGBUSCLK 0100: 4xGBUSCLK 0101: 5xGBUSCLK 0110: 6xGBUSCLK 0111: 7xGBUSCLK 1000: 8xGBUSCLK 1001: 9xGBUSCLK 1010: 10xGBUSCLK 1011: 11xGBUSCLK 1100: 12xGBUSCLK 1101: 13xGBUSCLK 1110: 14xGBUSCLK 1111: 15xGBUSCLK
3:0	SPW	Strobe Pulse Width	Strobe Pulse Width (Default: 1111) Specifies the low pulse width of the ND_RE* signal and the ND_WE* signal. 0000: 1xGBUSCLK 0001: 2xGBUSCLK 0010: 3xGBUSCLK 0011: 4xGBUSCLK 0100: 5xGBUSCLK 0101: 6xGBUSCLK 0110: 7xGBUSCLK 0111: 8xGBUSCLK 1000: 9xGBUSCLK 1001: 10xGBUSCLK 1010: 11xGBUSCLK 1011: 12xGBUSCLK 1100: 13xGBUSCLK 1101: 14xGBUSCLK 1110: 15xGBUSCLK 1111: 16xGBUSCLK

Figure 20.4.6 NAND Flash Memory Strobe Pulse Width Register (NDFSR)

20.4.7 NAND Flash Memory Reset Register (NDFRSTR) 0x5030

Bit(s)	Mnemonic	Field Name	Description
63:1	—	Reserved	—
0	RST	Reset	Reset (Default: 0) Setting "1" to this bit resets NDFMC. After the reset (2 GBUSCLK cycles later), this bit is automatically cleared to "0". 0: Don't care 1: Reset

Figure 20.4.7 NAND Flash Memory Reset Register (NDFRSTR)

20.5 Timing diagram

20.5.1 Command cycles and address cycles

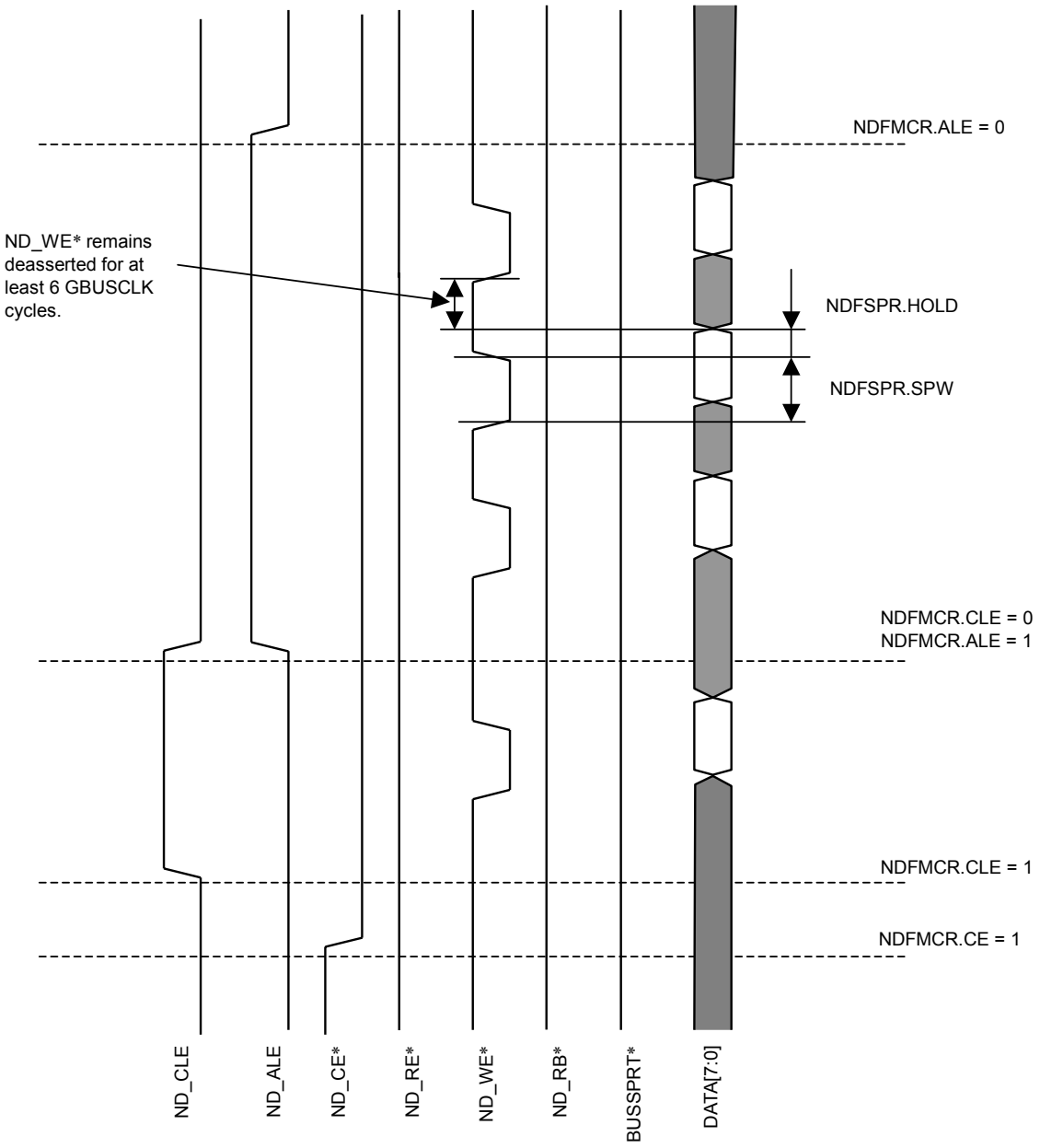


Figure 20.5.1 Command Cycles and Address Cycles

20.5.2 Data read cycles

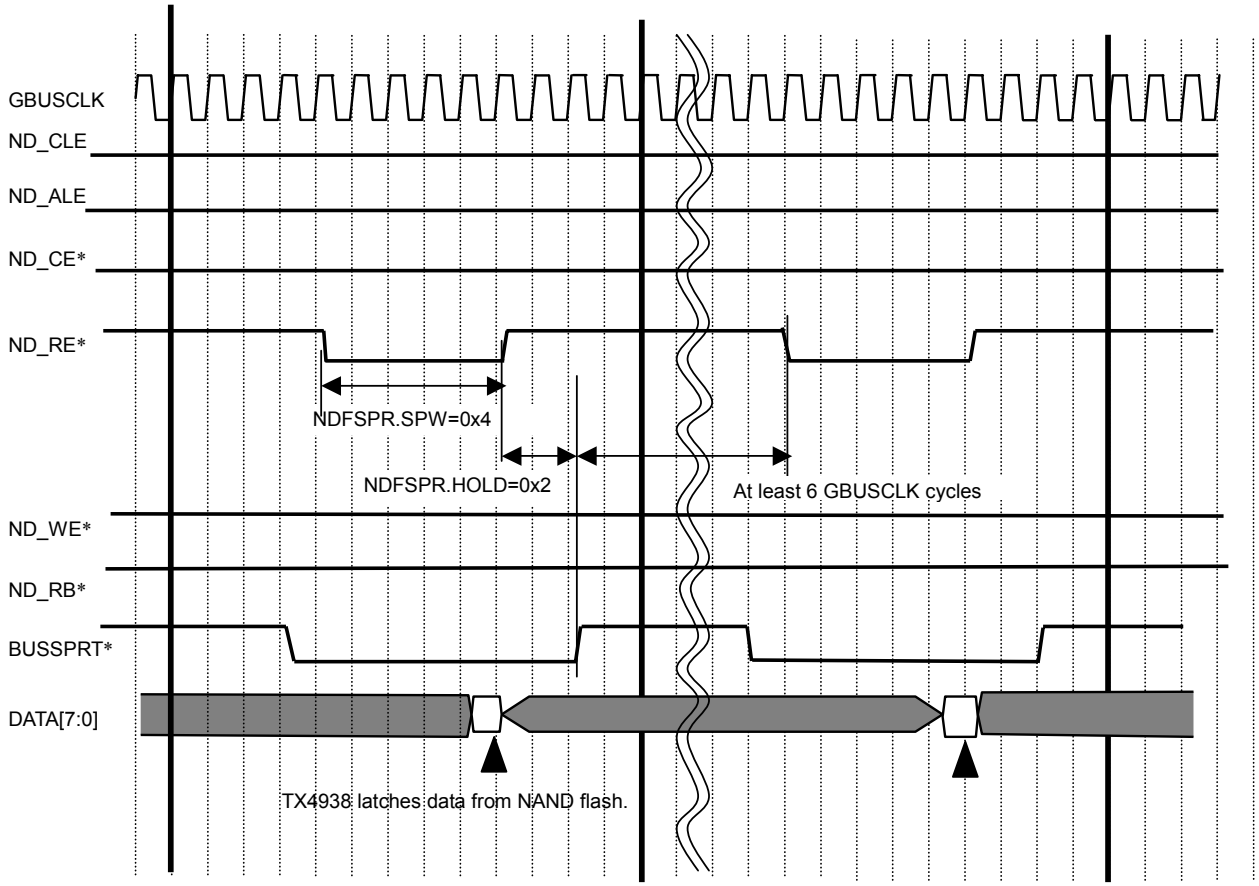


Figure 20.5.2 Data Read Cycles

20.5.3 Data write cycles

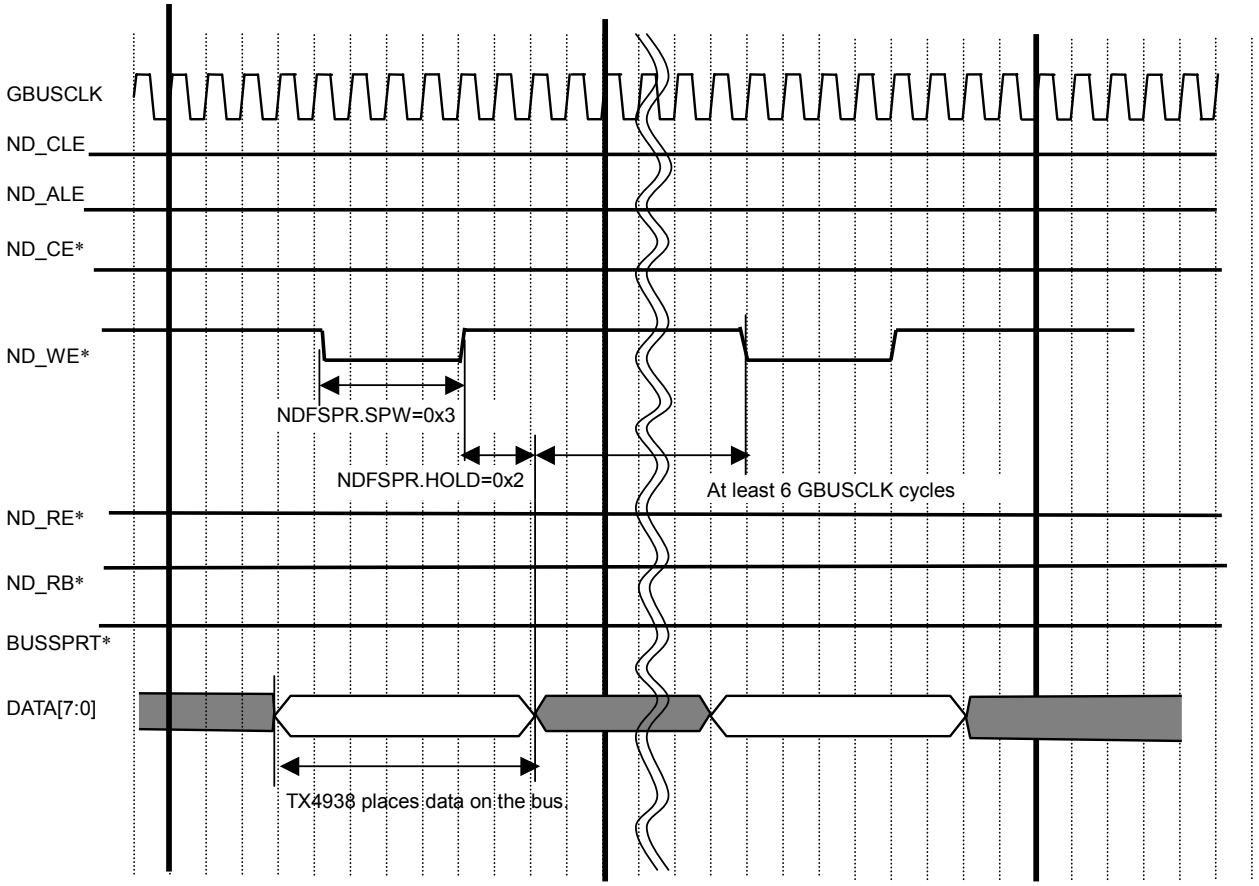


Figure 20.5.3 Data Write Cycles

20.6 NAND flash memory connection example

Figure 20.6.1 shows an example connection of NAND flash memory. BUSSPRT* is asserted when you read NAND flash memory. When connecting NAND flash memory to a Data Bus that performs bidirectional control using BUSSPRT*, connect to the TX4938's DATA via a buffer as shown below. When not performing bidirectional control using BUSSPRT*, directly connect to the TX4938's DATA.

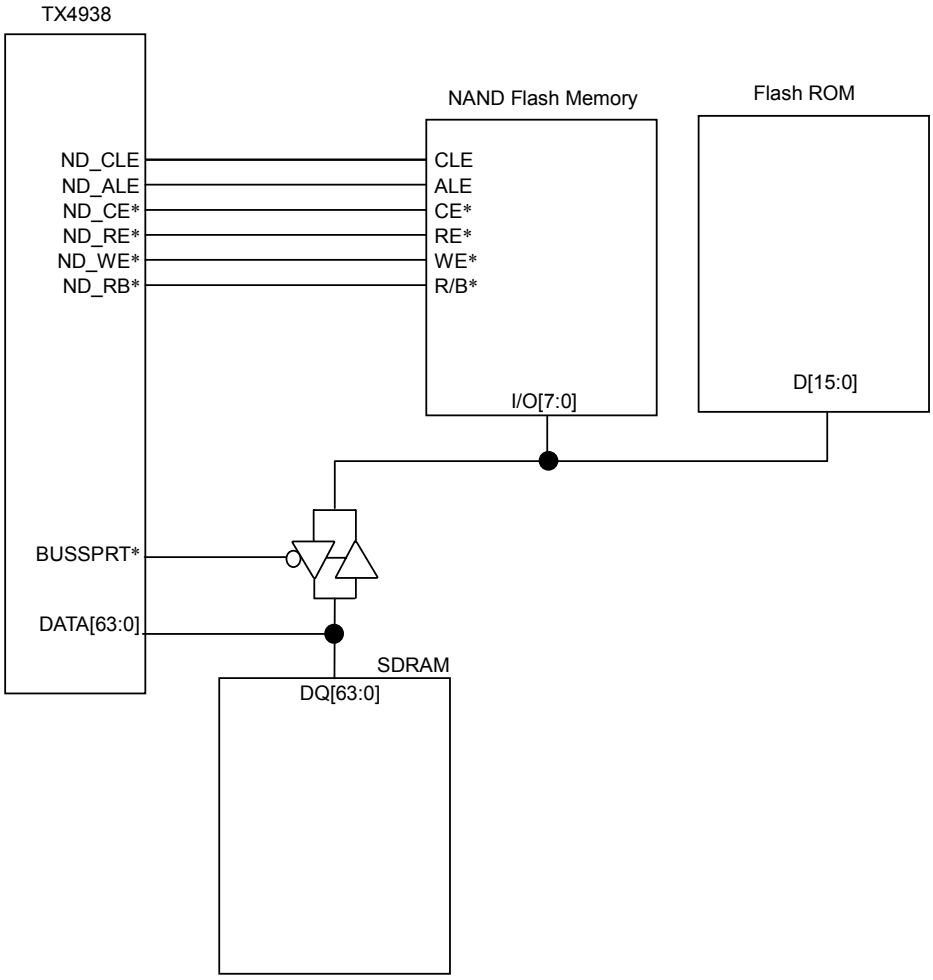


Figure 20.6.1 Connection Example of NAND Flash Memory

21. Extended EJTAG Interface

21.1 Extended EJTAG Interface

The TX4938 Extended EJTAG (Enhanced Joint Test Action Group) Interface provides two real-time debugging functions. One is the IEEE1149.1 standard compliant JTAG Boundary Scan Test, and the other is the Debugging Support Unit (DSU) that is built into the TX49/H3 core.

JTAG Boundary Scan Test

- IEEE1149.1 compatible TAP Controller
- Supports the following five instructions: EXTEST, SAMPLE/PRELOAD, IDCODE, BYPASS, HIGHZ

Real-time Debugging

- Real-time debugging using an emulation probe (made by Corelis or YDC)
- Execution control (run, break, step, register/memory access)
- Real-time PC tracing

Please contact your local Toshiba Sales representative for more information regarding how to connect the emulation probe.

The two functions of the Extended EJTAG Interface operate in one of two modes.

PC Trace Mode

- Execution control (fun, pause, access single steps, access internal register/system memory)
- JTAG Boundary Scan Test

Real-time Mode

- Real-time PC tracing

Refer to Section 3.1.11 for more information regarding signals used with the Extended EJTAG Interface.

Table 21.1.1 EJTAG Interface Function and Operation Code

PC Tracing Mode	Off	On
JTAG Boundary Scan	Boundary Scan Test	—
Real-time Debugging	Execution Control	Real-time PC Tracing

21.2 JTAG Boundary Scan Test

21.2.1 JTAG Controller and Register

The Extended EJTAG Interface contains a JTAG Controller (TAP Controller) and a Control Register. This section explains only those portions that are unique to the TX4938. Please refer to the "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture" for all other portion not covered here. Please contact your local Toshiba Sales representative for more information regarding the required BSDL files when performing the JTAG Boundary Scan Test.

- Instruction Register (Refer to 21.2.2)
- Data Register
  - Boundary Scan Register (Refer to 21.2.3)
  - Bypass Register
  - Device ID Register (Refer to 21.2.4)
  - JTAG Address Register
  - JTAG Data Register
  - JTAG Control Register
  - EJTAG Mount Register
- Test Access Port Controller (TAP Controller) (Refer to 21.3)

21.2.2 Instruction Register

The JTAG Instruction Register consists of an 8-bit shift register. This register is used for selecting either one or both of the test to be performed and the Test Data Register to be accessed. The Data Register is selected according to the instruction code in Table 21.2.1. Refer to the "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture" for more information regarding each instruction.

Table 21.2.1 Bit Configuration of JTAG Instruction Register

Instruction Code MSB → LSB	Instruction	Selected Data Register
00000000 (0x00)	EXTEST	Boundary Scan Register
00000001 (0x01)	SAMPLE/PRELOAD	Boundary Scan Register
00000010 (0x02)	Reserved	Reserved
00000011 (0x03)	IDCODE	Device ID Register
00000100 - 00001111	Reserved	Reserved
00010000 (0x10)	HIGHZ	Bypass Register
00010001 - 01111111	Reserved	Reserved
10000000 - 11111110	Refer to the "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture"	
11111111 (0xFF)	BYPASS	Bypass Register

Figure 21.2.1 shows the format of the Instruction Register.

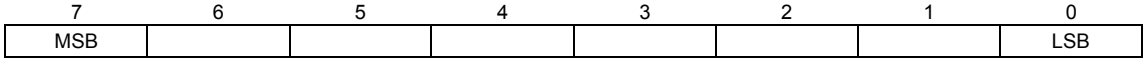


Figure 21.2.1 Instruction Register

The instruction code is shifted to the Instruction Register starting from the Least Significant Bit.

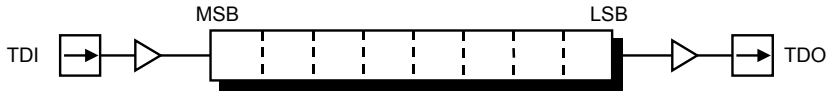


Figure 21.2.2 Shift Direction of the Instruction Register

21.2.3 Boundary Scan Register

The Boundary Scan Register contains a single 391-bit shift register to which all TX4938 I/O signals except for power supply, TDI, TCK, TDO, TMS, TRST*, and TEST[4]* are connected. Figure 21.2.3 shows the bits of the Boundary Scan Register.

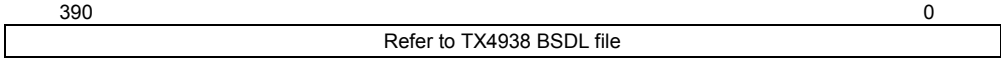


Figure 21.2.3 Boundary Scan Register

TDI input is fetched to the Most Significant Bit (MSB) of the Boundary Scan Register and the Least Significant Bit (LSB) of the Boundary Scan Register is sent from the TDO output.

Table 21.2.2 shows the boundary scan sequence relative to the processor signals.

Table 21.2.2 TX4938 Processor JTAG Scan Sequence (1/2)

JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name
	TDI	43	EEPROM_CS	86	PCIAD[20]
1	PIO[2]	44	PCST[3]	87	PCIAD[19]
2	PIO[1]	45	PCST[2]	88	PCIAD[26]
3	BYPASSPLL*	46	PCST[1]	89	PCIAD[31]
4	SD[1]	47	PCST[0]	90	PCIAD[29]
5	ACK*	48	PCIAD[0]	91	PCIAD[23]
6	BUSSPRT*	49	PCIAD[2]	92	PCIAD[25]
7	PIO[0]	50	PCIAD[1]	93	PCIAD[24]
8	SWE*	51	PCIAD[3]	94	PCIAD[30]
9	ACE*	52	PCIAD[5]	95	PCICLK[0]
10	CE[6]*	53	PCIAD[6]	96	GNT[0]*
11	CE[7]*	54	PCIAD[4]	97	REQ[1]*
12	CE[0]*	55	PCIAD[7]	98	GNT[1]*
13	CE[2]*	56	C_BE[0]	99	PCICLK[1]
14	CE[5]*	57	PCIAD[8]	100	REQ[0]*
15	CE[1]*	58	PCIAD[9]	101	PCICLK[2]
16	CE[3]*	59	PCIAD[11]	102	REQ[2]*
17	DMADONE*	60	PCIAD[12]	103	GNT[2]*
18	DMAREQ[3]	61	PCIAD[15]	104	PCICLK[3]
19	CE[4]*	62	M66EN	105	REQ[3]*
20	DMAACK[3]	63	PCIAD[13]	106	GNT[3]*
21	DMAACK[2]	64	PCIAD[10]	107	PME*
22	DMAREQ[2]	65	C_BE[1]	108	PCICLK[4]
23	DMAACK[1]	66	PAR	109	DATA[63]
24	DMAREQ[1]	67	PERR*	110	PCICLK[5]
25	DMAREQ[0]	68	PCIAD[14]	111	PCICLKIN
26	DMAACK[0]	69	SERR*	112	CGRESET*
27	BWE[3]*	70	LOCK*	113	MASTERCLK
28	BWE[0]*	71	STOP*	114	DATA[31]
29	BWE[2]*	72	IRDY*	115	DATA[62]
30	BWE[1]*	73	DEVSEL*	116	DATA[30]
31	EEPROM_DI	74	FRAME*	117	DATA[61]
32	DCLK	75	C_BE[2]	118	DATA[29]
33	TPC[1]	76	PCIAD[22]	119	DATA[60]
34	TPC[2]	77	TRDY*	120	DATA[59]
35	TPC[3]	78	PCIAD[17]	121	DATA[27]
36	EEPROM_DO	79	PCIAD[16]	122	DATA[28]
37	EEPROM_SK	80	C_BE[3]	123	DATA[58]
38	PCST[6]	81	PCIAD[28]	124	DATA[26]
39	PCST[7]	82	PCIAD[21]	125	DATA[57]
40	PCST[8]	83	PCIAD[18]	126	DATA[56]
41	PCST[5]	84	PCIAD[27]	127	DATA[24]
42	PCST[4]	85	ID_SEL	128	DATA[25]

Table 21.2.2 TX4938 Processor JTAG Scan Sequence (2/2)

JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name
129	DATA[55]	172	ADDR[8]	215	DATA[38]
130	DATA[54]	173	ADDR[7]	216	DATA[36]
131	DATA[22]	174	ADDR[6]	217	DATA[5]
132	DATA[53]	175	ADDR[5]	218	DATA[34]
133	DATA[23]	176	ADDR[4]	219	DATA[3]
134	DATA[21]	177	ADDR[2]	220	DATA[33]
135	DATA[52]	178	ADDR[1]	221	DATA[35]
136	DATA[50]	179	ADDR[3]	222	DATA[2]
137	DATA[20]	180	ADDR[0]	223	DATA[1]
138	DATA[48]	181	RAS*	224	DATA[0]
139	CB[3]	182	SDCS[1]*	225	DATA[32]
140	DATA[51]	183	DQM[5]	226	WDRST*
141	DATA[19]	184	DQM[1]	227	OE*
142	DQM[7]	185	SDCS[0]*	228	SYSCLK
143	DATA[17]	186	DQM[4]	229	TEST[1]*
144	CB[2]	187	WE*	230	RESET*
145	CB[7]	188	CAS*	231	TEST[0]*
146	DATA[49]	189	CB[5]	232	SCLK
147	DATA[18]	190	DQM[0]	233	HALTDOZE
148	SDCS[3]*	191	CB[1]	234	TXD[1]
149	DQM[2]	192	CB[4]	235	TXD[0]
150	DATA[16]	193	DATA[47]	236	RTS[1]*
151	CB[6]	194	CB[0]	237	RTS[0]*
152	DQM[6]	195	DATA[12]	238	CTS[1]*
153	DQM[3]	196	DATA[42]	239	RXD[1]
154	SDCLK[1]	197	DATA[46]	240	CTS[0]*
155	ADDR[17]	198	DATA[15]	241	INT[5]
156	CKE	199	DATA[40]	242	INT[4]
157	SDCS[2]*	200	DATA[44]	243	INT[3]
158	SDCLK[3]	201	DATA[14]	244	INT[2]
159	ADDR[18]	202	DATA[9]	245	INT[1]
160	ADDR[19]	203	DATA[13]	246	TCLK
161	ADDR[15]	204	DATA[45]	247	RXD[0]
162	ADDR[16]	205	DATA[11]	248	INT[0]
163	ADDR[14]	206	DATA[7]	249	NMI*
164	SDCLKIN	207	DATA[43]	250	TIMER[0]
165	SDCLK[0]	208	DATA[41]	251	PIO[7]
166	SDCLK[2]	209	DATA[10]	252	TIMER[1]
167	ADDR[12]	210	DATA[39]	253	PIO[6]
168	ADDR[13]	211	DATA[8]	254	PIO[5]
169	ADDR[10]	212	DATA[4]	255	PIO[4]
170	ADDR[11]	213	DATA[37]	256	PIO[3]
171	ADDR[9]	214	DATA[6]		TDO

21.2.4 Device ID Register

The Device ID Register is a 32-bit shift register. This register is used for reading the ID code that expresses the IC manufacturer, part number, and version from the IC and sending it to a serial device. The following figure shows the configuration of the Device ID Register.

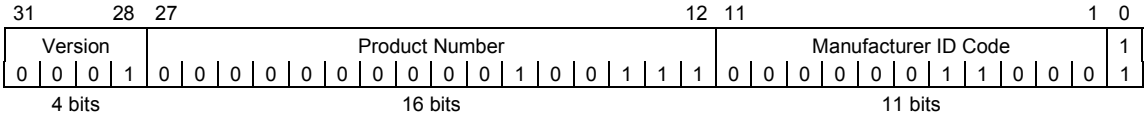


Figure 21.2.4 Device ID Register

The device ID code for the TX4938 is 0x10027031. However, the four top bits of the Version field may be changed. The device ID code is shifted out from the Least Significant Bit.

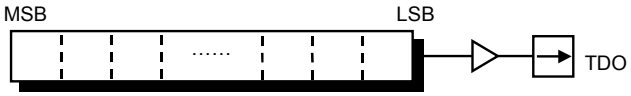


Figure 21.2.5 Shift Direction of the Device ID Register

21.3 Initializing the Extended EJTAG Interface

The Extended EJTAG Interface is not reset by asserting the RESET* signal. Operation of the TX49/H3 core is not guaranteed if the Extended EJTAG Interface is not reset. This interface is initialized by either of the following methods.

- Assert the TRST* signal.
  (TRST* signal is pulled down (by ex. 10 kΩ))
- After clearing the processor reset, set the TMS input to High for five consecutive rising edges of the TCK input. The reset state is maintained if TMS is able to maintain the High state.

The above methods must be performed while the MASTERCLK signal is being input.

The G-Bus Time Out Detection function is disabled when the TRST* signal is deasserted. (Refer to Section 5.1.1.)

22. Electrical Characteristics

22.1 Absolute maximum rating (*1)

Item	Symbol	Rating	Unit
Supply Voltage (for I/O pins)	$V_{CCIOMax}$	−0.3 - 3.9	V
Supply Voltage (for internal circuits)	$V_{CCIntMax}$	−0.3 - 3.0	V
Input Voltage (*2)	V_{IN}	−0.3 - V_{CCIO} + 0.3V	V
Storage Temperature	T_{STG}	−40 - +125	°C

(*1) The absolute maximum rating is a rating that must not be exceeded even for an instant even by a single item. Exceeding the absolute maximum rating may cause component damage or degradation, and may result in damage or combustion that causes bodily harm. When designing application devices, be absolutely sure that this absolute maximum rating is never exceeded.

(*2) Even with VCCIO + 0.3 V, be sure to never exceed the VCCIOMax maximum rating.

22.2 Recommended operating conditions

Item		Symbol	Condition	Min.	Max.	Unit
Supply Voltage	I/O	V_{CCIO}		3.1	3.5	V
	Internal Circuit	V_{CCInt}		1.4	1.6	V
Operating Temperature (Package Temperature)		T_C		0	70	°C

(*3) A recommended operating condition is a usage condition that Toshiba recommends for a product to function properly and maintain a uniform level of quality. Using a product such that even one item is not under the recommended operating conditions may cause a malfunction to occur. When designing application devices, be sure that these recommended operating conditions ranges are never exceeded.

22.3 DC characteristics

22.3.1 DC characteristics of pins other than PCI Interface pins

(Tc = 0 - 70°C, V_{ddIO} = 3.3V ± 0.2V, V_{CCInt} = 1.5V ± 0.1V, V_{SS} = 0V)

Item	Symbol	Condition	Min.	Max.	Unit
Low-level Input Voltage	V_{IL1}	(1)	−0.3	0.8	V
High-level Input Voltage	V_{IH1}	(1)	2.0	V_{CCIO} + 0.3	V
Low-level Output Current	I_{OL1} I_{OL2}	(2)V_{OL} = 0.4 V (3)V_{OL} = 0.4 V	8 4	— —	mA mA
Low-level Output Current	I_{OL3} I_{OL4}	(4)V_{OL} = 0.4 V (5)V_{OL} = 0.4 V	16 8	— —	mA mA
High-level Output Current	I_{OH1} I_{OH2}	(2)V_{OH} = 2.4 V (3)V_{OH} = 2.4 V	— —	−8 −4	mA mA
High-level Output Current	I_{OH3} I_{OH4}	(4)V_{OH} = 2.4 V (5)V_{OH} = 2.4 V	— —	−16 −8	mA mA
Low-level Input Leak Current	I_{IL1} I_{IL2}	(6)V_{IN}=V_{SS} (7)V_{IN}=V_{SS}	−10 −200	10 −10	μA μA
High-level Input Leak Current	I_{IH1}	(8)V_{IN}=V_{CCIO}	−10	10	μA
Hi-Z Output Leak Current	I_{OZ}	(9)	−10	10	μA
Operating Current (Internal)	I_{CCInt}	V_{ddIN} = 1.6 V, Internal core frequency = 300 MHz	—	500	mA
Operating Current (I/O Pin)	I_{CCIO}	V_{ddIO} = 3.5 V, External bus frequency = 100 MHz Pin capacitance load = 25 pF	—	400	mA

(1) : All input pins except for the PCI interface signal pins, and all bidirectional pins (during input)

(2) : ACE*, ACK*, BUSSPRT*, BWE[3:0]*, CE[7:0]*, DMAACK[3:0], DMADONE*, WDRST*, HALTDOZE, PIO[7:0], RTS[1:0], SWE*, SYSCLK, TIMER[1:0] and TXD[1:0]

(3) : DCLK, PCST[8:0], TDO and TPC[3:1]

(4) : ADDR[19:0], CAS*, CB[7:0], CKE, DATA[63:0], DQM[7:0], OE*, RAS*, SDCLK[3:0], SDCLKIN, SDCS[3:0]*, and WE* when drive capacity is set to 16 mA.

(5) : ADDR[19:0], CAS*, CB[7:0], CKE, DATA[63:0], DQM[7:0], OE*, RAS*, SDCLK[3:0], SDCLKIN, SDCS[3:0]*, and WE* when drive capacity is set to 8 mA.

(6) : CGRESET*, RESET*, TRST*, BYPASSPLL*, MASTERCLK, SDCLKIN and SDIN[1]

(7) : CTS[1:0]*, DMAREQ[3:0], DMADONE*, RXD[1:0], SCLK, TCLK, INT[5:0], TCK, TDI, TEST[4:0]*, TMS, ACK*, CB[7:0], DATA[63:0], ADDR[19:0], NMI* and PIO[7:0]

(8) : Signal in (6) and (7) above

(9) : DCLK, TDO

22.3.2 DC characteristics of PCI Interface pins

(Tc = 0 – 70°C, V_{ddIO} = 3.3 V ± 0.2 V, V_{ddIN} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
Low-level Input Voltage	V_{ILPCI}	(1)	−0.5	0.9	V
High-level Input Voltage	V_{IHPCI}	(1)	1.8	VddIO + 0.3	V
High-level Output Voltage	V_{OHPCI}	(2) I_{OUT} = −500 μA	$V_{ddIO} \times 0.9$	—	V
Low-level Output Voltage	V_{OLPCI}	(2) I_{OUT} = 1500 μA	—	VddIO × 0.1	V
High-level Output Current	I_{OH1}	(4) V_{OH} = 2.4 V	—	-8	mA
Low-level Output Current	I_{OL1}	(4) V_{OL} = 0.4 V	8	—	mA
Input Leak Current	I_{IHPCI} I_{ILPCI}	0 < V_{IN} < V_{ddIO}	−10 −10	10 10	μA μA
High-level Input Leak Current	I_{IL1}	(5) V_{IN} = V_{ddIO}	−10	10	μA
Low-level Input Leak Current	I_{IL2}	(5) V_{IN} = VSS	−200	−10	μA
Hi-Z Output Leak Current	I_{OZPCI}	(3)	−10	10	μA

(1) : ID_SEL, PCICLKIN, C_BE[3:0], DEVSEL*, FRAME*, GNT[3:0]*, IRDY*, LOCK*, M66EN, PAR, PCIAD[31:0], PERR*, PME*, REQ[3:0], SERR*, STOP*, TRDY*

(2) : All PCI interface signals except for ID_SEL, LOCK*, PCICLKIN

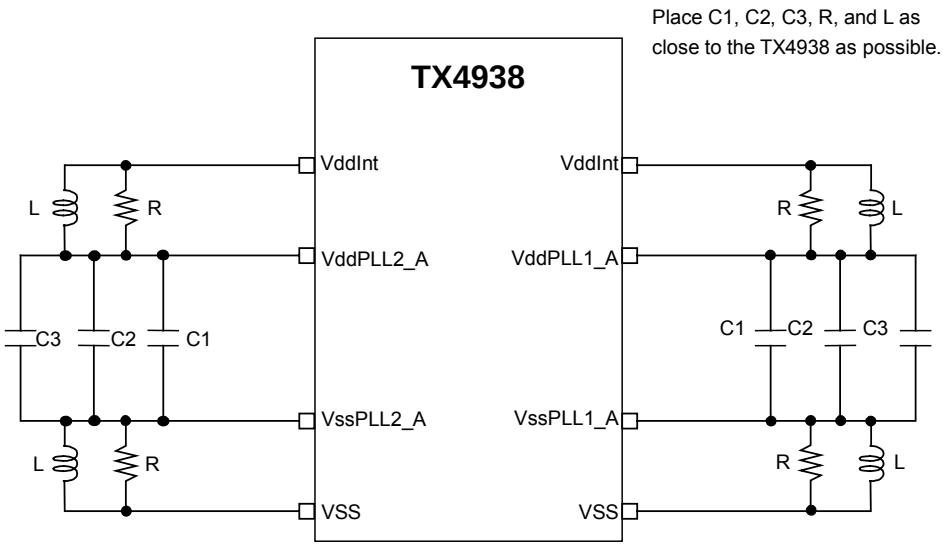
(3) : PME*, REQ[1], SERR*

(4) : EEPROM_CS, EEPROM_DO, EEPROM_SK

(5) : EEPROM_DI

22.4 PLL power

22.4.1 PLL power connection example



Item	Symbol	Recommended Value	Unit
Resistance	R	5.6	Ω
Inductance	L	2.2	μH
Condensor (capacitance)	C1	1	NF
	C2	82	NF
	C3	10	μF
VddInt / VddPLL		1.5 ± 0.1	V

Note that the above values are recommended values.

Figure 22.4.1 Oscillation Circuit

22.5 AC characteristics

22.5.1 MASTERCLK AC characteristics

($Tc = 0 – 70°C$, $V_{CCIO} = 3.3\ V \pm 0.2\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, VSS = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
MASTERCLK Cycle	t_{MCP}	Boot configuration ADDR[2]=H	7.5	80	ns
		Boot configuration ADDR[2]=L	30	320	ns
MASTERCLK Frequency *1)	f_{MCK}	Boot configuration ADDR[2]=H	12.5	133.3	MHz
		Boot Configuration ADDR[2]=L	3.125	33.3	MHz
MASTERCLK High Time	t_{MCH}		2	—	ns
MASTERCLK Low Time	t_{MCL}		2	—	ns
CPUCLK Frequency	f_{CPU}	TMPR4938XBG-300	50	300	MHz
		TMPR4938XBG-333	50	333	
MASTERCLK Rise Time	t_{MCR}		—	1.5	ns
MASTERCLK Fall Time	t_{MCF}		—	1.5	ns

*1) TX4938 operation is only guaranteed when the power is stable, PLL secures the PLL oscillation stability time tMCP_PLL and is in the Enable state.

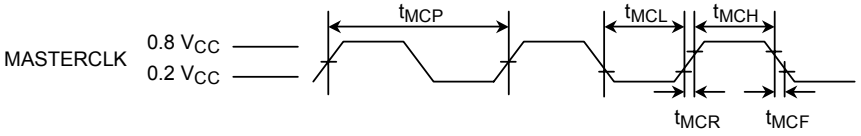
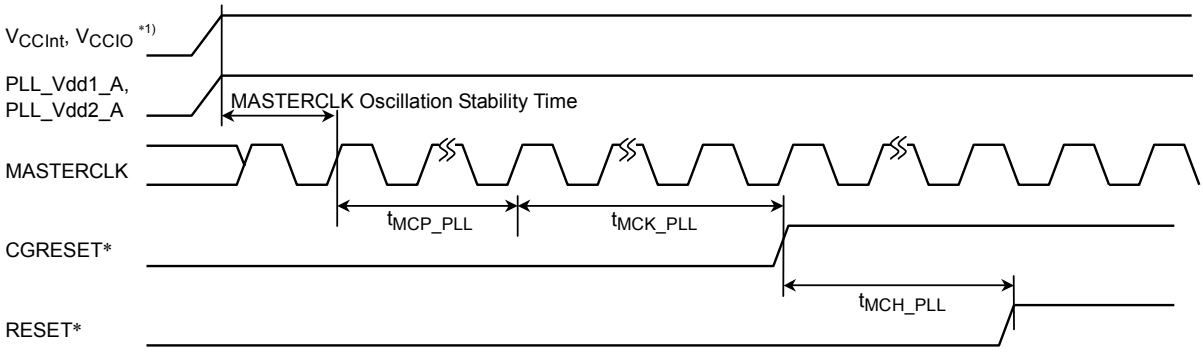


Figure 22.5.1 Timing Diagram: MASTERCLK

22.5.2 Power on AC characteristics

($Tc = 0 – 70°C$, $V_{CCIO} = 3.3\ V \pm 0.2\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, VSS = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
PLL Oscillation Stability Time	t_{MCP_PLL}		10		ms
CGRESET* Width Time	f_{MCK_PLL}		1		ms
RESET* Width Time	t_{MCH_PLL}		1	—	ms



*1) V_{CCInt} and V_{CCIO} must start up simultaneously, or V_{CCInt} must be first.
The difference of the stand up time of a power supply within in 100 m seconds.

Figure 22.5.2 Timing Diagram: Power On Reset

22.5.3 SDRAM Interface AC characteristics

(Tc = 0 – 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, VSS = 0 V, CL = 150 pF)

Item	Symbol	Conditions	Min.	Max.	Unit
SDCLK[3:0] Cycle Time	t_{CYC_SDCLK}	C_L=50 pF, 16 mA buffer time	7.5	—	ns
SDCLK[3:0] High Time	t_{HIGH_SDCLK}	C_L=50 pF, 16 mA buffer time	2.5	—	ns
SDCLK[3:0] Low Time	t_{LOW_SDCLK}	C_L=50 pF, 16 mA buffer time	2.5	—	ns
SDCLKIN Input Skew	t_{BP}	When in the Non-bypass mode *4)	0	t_{CYC_SDCLK} – 5.5	ns
ADDR[19:5] Output Delay	t_{VAL_ADDR1}	C_L=150 pF, 16 mA buffer, *1)	1.5	6.5	ns
		C_L= 80 pF, 16 mA buffer, *1)	1.5	5.2	ns
SDCS[3:0]* Output Delay	t_{VAL_SDCS}	C_L= 80 pF, 16 mA buffer	1.5	5.2	ns
RAS* Output Delay	t_{VAL_RAS}	C_L=150 pF, 16 mA buffer, *1)	1.5	6.5	ns
		C_L= 80 pF, 16 mA buffer, *1)	1.5	5.2	ns
CAS* Output Delay	t_{VAL_CAS}	C_L=150 pF, 16 mA buffer, *3)	1.5	6.5	ns
		C_L= 80 pF, 16 mA buffer, *3)	1.5	5.2	ns
WE* Output Delay	t_{VAL_WE}	C_L=150 pF, 16 mA buffer, *3)	1.5	6.5	ns
		C_L= 80 pF, 16 mA buffer, *3)	1.5	5.2	ns
CKE Output Delay	t_{VAL_CKE}	C_L=150 pF, 16 mA buffer	1.5	6.5	ns
		C_L= 80 pF, 16 mA buffer	1.5	5.2	ns
DQM[7:0] Output Delay	t_{VAL_DQM}	C_L= 50 pF, 16 mA buffer, *2)	1.5	6.5	ns
		C_L= 30 pF, 16 mA buffer, *2)	1.5	5.2	ns
DATA[63:0] Output Delay (H→L, L→H)	t_{VAL_DATA1}	C_L= 50 pF, 16 mA buffer, *2)	1.5	6.5	ns
		C_L= 30 pF, 16 mA buffer, *2)	1.5	5.2	ns
DATA[63:0] Output Delay (Valid → Hi-Z)	$t_{VAL\_DATA1ZV}$	C_L= 50 pF, 16 mA buffer, *2)	1.5	6.5	ns
		C_L= 30 pF, 16 mA buffer, *2)	1.5	5.2	ns
DATA[63:0] Output Delay (Valid→Hi-Z)	$t_{VAL\_DATA1VZ}$	C_L= 50 pF, 16 mA buffer, *2)	1.5	6.5	ns
		C_L= 30 pF, 16 mA buffer, *2)	1.5	5.2	ns
DATA[63:0] Input Setup Time	t_{SU_DATA1B}	When in the Bypass mode	4.0	—	ns
DATA[63:0] Input Hold Time	t_{HO_DATA1B}	When in the Bypass mode	0.5	—	ns
DATA[63:0] Input Setup Time	$t_{SU\_DATA1NB}$	When in the Non-bypass mode	1.0	—	ns
DATA[63:0] Input Hold Time	$t_{HO\_DATA1NB}$	When in the Non bypass mode	1.0	—	ns

*1) Becomes a 2-cycle signal when tDACT of SDCTR1 is “1”.

*2) Becomes a 2-cycle signal when tSWB of SDCTR1 is “1”.

*3) 2-cycle signal

For information on 2-cycle operation, see the description in Chapter 9 SDRAM Controller.

*4) The MAX value is is t_{CYC_SDCLK} – 5.5 ns.

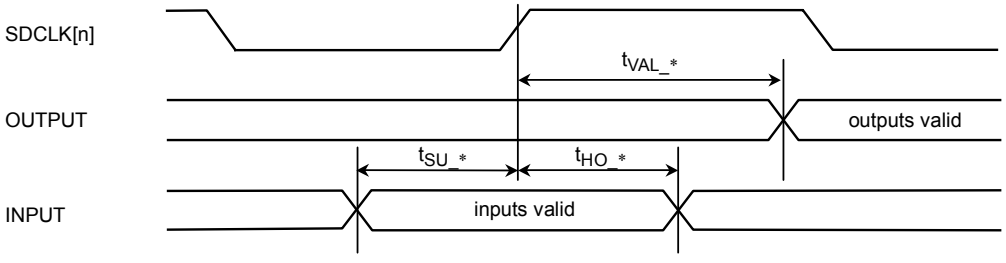


Figure 22.5.3 Timing Diagram: Output Signal and Input Signal when in the Bypass Mode (SDCLK Reference)

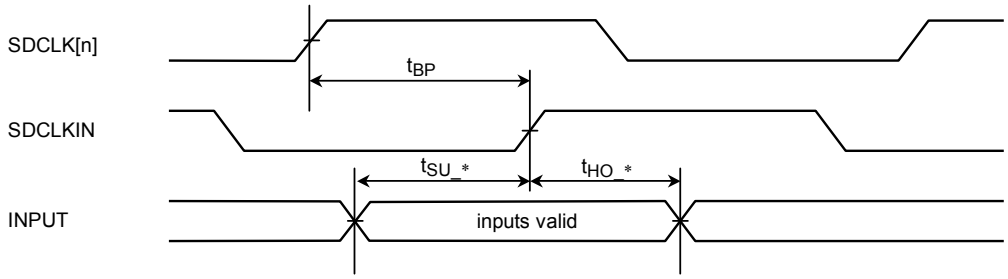


Figure 22.5.4 Timing Diagram: Input Signal when in the Non-bypass Mode (SDCLK Reference)

22.5.4 External Bus Interface AC characteristics

($Tc = 0 – 70°C$, $V_{CCIO} = 3.3\ V \pm 0.2\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Item	Symbol	Condition	Min.	Max.	Unit
SYSCLK Cycle Time	t_{CYC_SYSCLK}	Buffer fixed: C_L=50 pF, 8 mA	7.5	—	ns
SYSCLK High Time	t_{HIGH_SYSCLK}	Buffer fixed: C_L=50 pF, 8 mA	4	—	ns
SYSCLK Low Time	t_{LOW_SYSCLK}	Buffer fixed: C_L=50 pF, 8 mA	4	—	ns
ADDR[19:5] Output Delay	t_{VAL_ADDR2}	For C_L=150 pF, 16 mA buffer	1.5	6.5	ns
CE[7:0]* Output Delay	t_{VAL_CE}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
OE* Output Delay	t_{VAL_OE}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
SWE* Output Delay	t_{VAL_SWE}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
BWE[3:0]* Output Delay	t_{VAL_BWE}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
ACE Output Delay	t_{VAL_ACE}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
BUSSPRT* Output Delay	t_{VAL_DQM}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
DATA[63:0] Output Delay (H→L, L→H)	t_{VAL_BUS}	For C_L=50 pF, 16 mA buffer	1.5	6.5 *1)	ns
DATA[31:0] Output Delay (Hi-Z→Valid)	$t_{VAL\_DATA2ZV}$	For C_L=50 pF, 16 mA buffer	1.5	8.5	ns
DATA[31:0] Ouput Delay (Valid→Hi-Z)	$t_{VAL\_DATA2VZ}$	For C_L=50 pF, 16 mA buffer	1.5	8.5	ns
DATA[31:0] Input Setup Time	t_{SU_DATA2}		6.0	—	ns
DATA[31:0] Input Hold Time	t_{HO_DATA2}		0.5	—	ns
ACK* Output Delay (H→L, L→H)	t_{VAL_ACK}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
ACK* Output Delay (Hi-Z→Valid)	t_{VAL_ACKZV}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
ACK* Output Delay (Valid→Hi-Z)	t_{VAL_ACKVZ}	Buffer fixed: C_L=50 pF, 8 mA	1.5	8.5	ns
ACK* Input Setup Time	t_{SU_ACK}		6.0	—	ns
ACK* Input Hold Time	t_{HO_ACK}		0.5	—	ns

*1) When the speed of the External Bus is set to 1/3 speed, the delay becomes the GBUSCLK cycle + 6.5 ns. EBCCRn.SP of the External Bus Controller sets the speed of the External Bus.

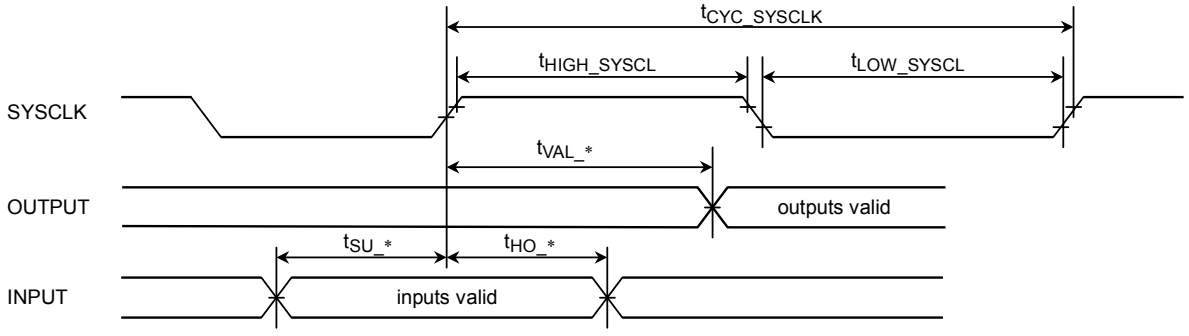


Figure 22.5.5 Timing Diagram: External Bus Interface

22.5.5 PCI Interface AC characteristics (66 MHz)

(Tc = 0 – 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
PCICLKIN Cycle Time (66 MHz)	t_{CYC66}		15	30	ns
PCICLKIN High Time (66 MHz)	t_{HIGH66}		6	—	ns
PCICLKIN Low Time (66 MHz)	t_{LOW66}		6	—	ns
PCICLKIN Slew Rate (66 MHz)	t_{SLEW66}		1.5	4	V/ns
PCICLK[5:0] Cycle Time (66 MHz)	t_{CYCO66}	C_L=50 pF	15	30	ns
PCICLK[5:0] High Time (66 MHz)	$t_{HIGHO66}$	C_L=50 pF	6	—	ns
PCICLK[5:0] Low Time (66 MHz)	t_{LOWO66}	C_L=50 pF	6	—	ns
PCICLK[5:0] Skew (66 MHz)	t_{SKEW}	C_L=50 pF, point to point connect	0	1	ns
PCI Output Signal*1) Output Delay	t_{VAL66}	C_L=30 pF	2	8	ns
PCI Input Signal*2) Input Setup Time	t_{SU66}		3	—	ns
PCI Input Signal*2) Input Hold Time	t_{HO66}		0.5	—	ns
ID_SEL, REQ[0]*, GNT[3:0]* Output Delay	$t_{VALPP66}$	C_L=30 pF, point-to-point connect	2	8	ns
ID_SEL, REQ[3:0]*, GNT[0]* Input Setup Time	t_{SUPP66}	Point-to-point connection	5	—	ns
ID_SEL, REQ[3:0]*, GNT[0]* Input Hold Time	t_{HOPP66}	Point-to-point connection	0	—	ns

1) PCIAD[31:0], C_BE[3:0], PAR, FRAME, IRDY*, TRDY*, STOP*, DEVSEL*, PERR*, SERR*, M66EN, and PME*

2) PCIAD[31:0] , C_BE[3:0] , PAR, FRAME, IRDY*, TRDY*, STOP*, DEVSEL*, PERR*, SERR*, M66EN, PME*, LOCK*, and ID_SEL

22.5.6 PCI Interface AC characteristics (33 MHz)

(Tc = 0 – 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Condition	Min.	Max.	Unit
PCICLKIN Cycle Time (33 MHz)	t_{CYC33}		30	40	ns
PCICLKIN High Time (33 MHz)	t_{HIGH33}		11	—	ns
PCICLKIN Low Time (33 MHz)	t_{LOW33}		11	—	ns
PCICLKIN Slew Rate (33 MHz)	t_{SLEW33}		1	4	V/ns
PCICLK[5:0] Cycle Time (33 MHz)	t_{CYC33}	C_L=70 pF	30	40	ns
PCICLK[5:0] High Time (33 MHz)	t_{HIGH33}	C_L=70 pF	11	—	ns
PCICLK[5:0] Low Time (33 MHz)	t_{LOW33}	C_L=70 pF	11	—	ns
PCICLK[5:0] Skew (33 MHz)	t_{SKEW}	C_L=70 pF, point to point connect	0	2	ns
PCI Output Signal*1) Output Delay	t_{VAL33}	C_L=70 pF	2	11	ns
PCI Input Signal*2) Input Setup Time	t_{SU33}		7	—	ns
PCI Input Signal*2) Input Hold Time	t_{HO33}		0.5	—	ns
ID_SEL, REQ[0]*, GNT[3:0]* Output Delay	$t_{VALPP33}$	C_L=70 pF, point to point connect	2	12	ns
ID_SEL, REQ[3:0]*, GNT[0]* Input Setup Time	t_{SUPP33}	point to point connct	10	—	ns
ID_SEL, REQ[3:0]*, GNT[0]* Input Hold Time	t_{HOPP33}	point to point connect	0	—	ns

1) PCIAD[31:0], C_BE[3:0] , PAR, FRAME, IRDY*, TRDY*, STOP*, DEVSEL*, PERR*, SERR*, M66EN, and PME*

2) PCIAD[31:0] , C_BE[3:0] , PAR, FRAME, IRDY*, TRDY*, STOP*, DEVSEL*, PERR*, SERR*, M66EN, PME*, LOCK*, and ID_SEL

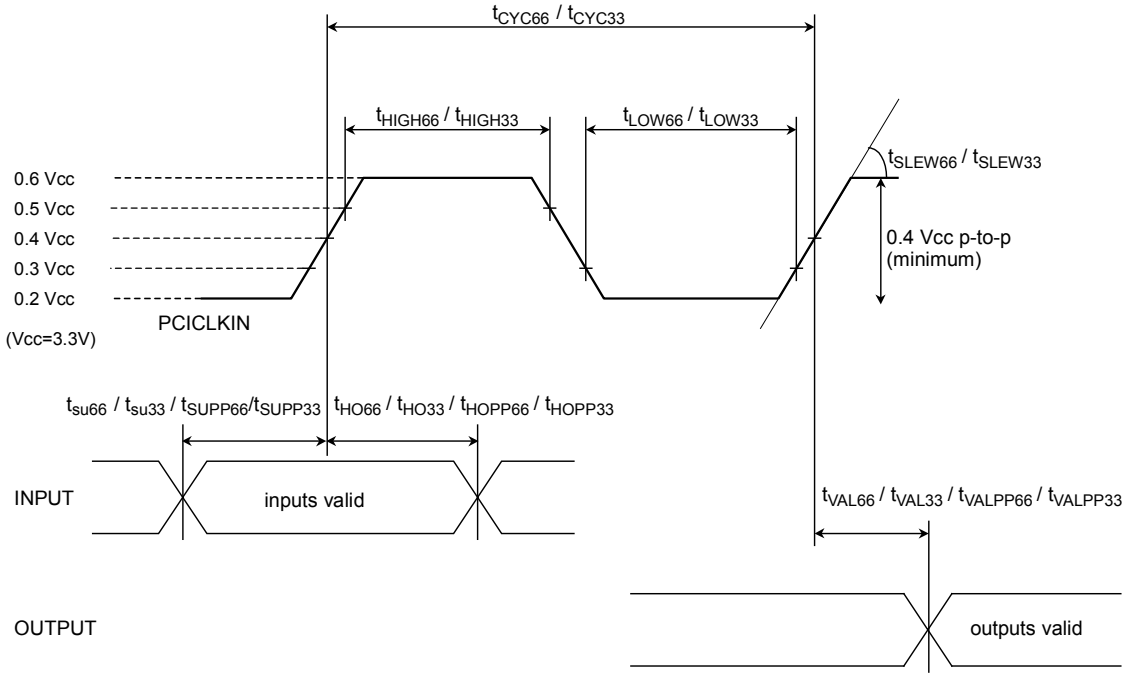


Figure 22.5.6 Timing Diagram: PCI Interface (3.3 V)

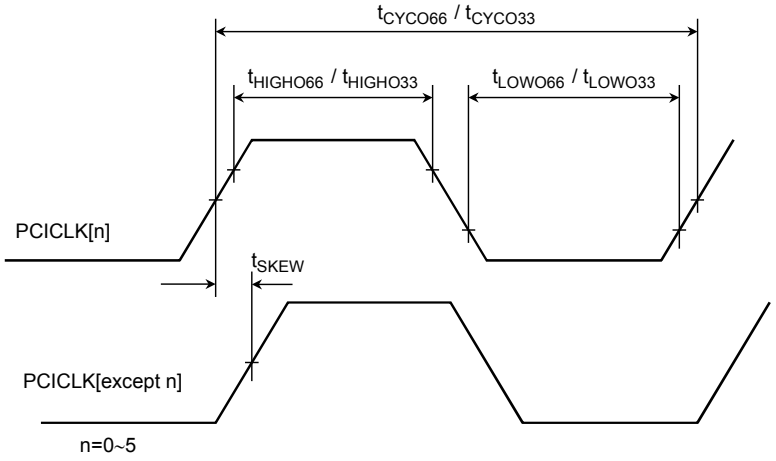


Figure 22.5.7 Timing Diagram: PCI Clock Skew

22.5.7 PCI EEPROM Interface AC characteristics

(T_c = 0 ~ 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
EEPROM_SK High Time	t_{HIGH_EPSK}	C_L=50 pF	500	—	ns
EEPROM_SK Low Time	t_{LOW_EPSK}	C_L=50 pF	500	—	ns
EEPROM_DO Output Delay Time *1)	t_{VAL_EPDO}	C_L=50 pF	—	100	ns
EEPROM_DI Input Setup Time	t_{SU_EPDI}		100	—	ns
EEPROM_DI Input Hold Time	t_{HO_EPDI}		100	—	ns
EEPROM_CS Output Delay Time	t_{VAL_CS}	C_L=50 pF	100	—	ns

*1) The TX4938 controls EEPROM and its control signal synchronous to the falling edge of EEPROM_SK. Since the EEPROM operates at the rising edge of EEPROM_SK, you do not have to take the MIN side of EEPROM_DO into account.

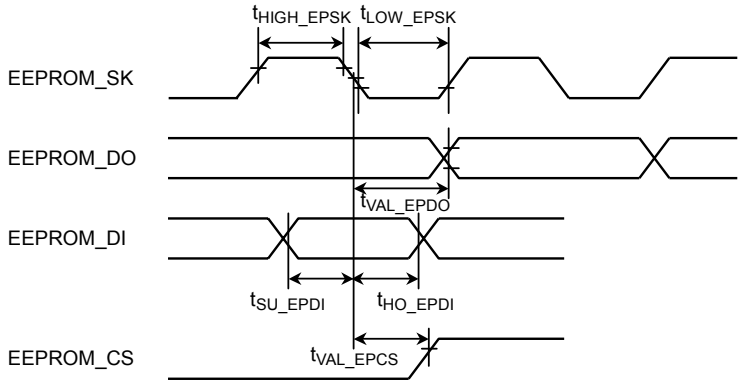


Figure 22.5.8 Timing Diagram: PCI EEPROM Interface

22.5.8 DMA Interface AC characteristics

(T_c = 0 – 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
DMADONE* Delay	t_{VAL_DONE}	C_L=50 pF SYSCLK (C_L=50 pF) reference	—	12	ns
DMADONE* Input Pulse Width Time	t_{PW_DONE}	Boot configuration ADDR[2]=H	t_{MCP} × 1.1	—	ns
		Boot configuration ADDR[2]=L	1/4 × t_{MCP} × 1.1	—	ns

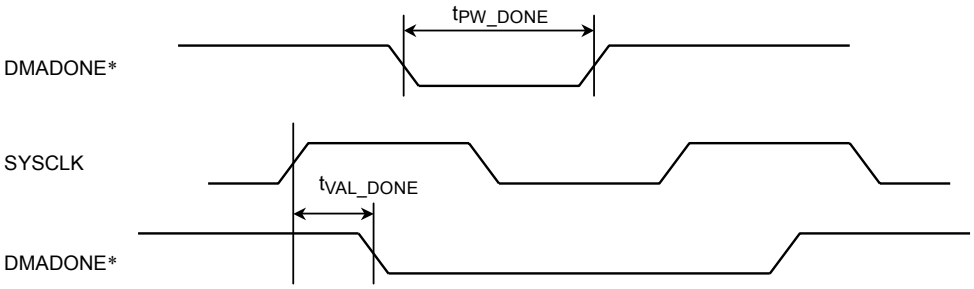


Figure 22.5.9 Timing Diagram: DMA Interface

Notes:

(1) DMAREQ[n]

Edge Detection: Set the pulse width to 1.1× the GBUSCLK cycle or higher.

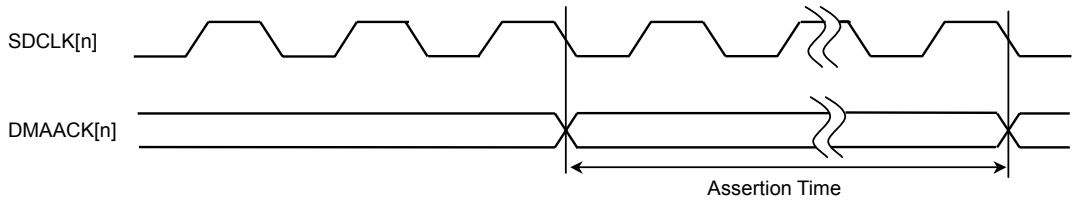
Level Detection: There is no AC characteristic definition. Continue asserting DMAREQ[3:0] until DMAACK[3:0] is received.

(2) DMAACK[n]

The DMAACK[n] signal is synchronous to SDCLK. (It is driven by GUBSCLK inside the chip. See Chapter 6 for more information.)

The DMAACK[n] signal is asserted by SYSCLK or SDCLK for 3 cycles or more. However, this is changed by the conditions [1] and [2] below.

[1] DMAC transfer mode (Single Address transfer, Dual Address transfer)
[2] Access time of the device DMAC accesses



- When driving an external device with SYSCLK
  Is asserted by SYSCLK for at least 3 cycles even in the shortest assertion case.
- When driving an external device with SDCLK
  Is asserted by SDCLK for at least 3 cycles even in the shortest assertion case. The AC characteristics for Single Address transfer with SDRAM are tight, so we do not recommend Single Address transfer.

(3) DMADONE*

Is asserted for only 1 SYSCLK cycle synchronous to SYSCLK.

22.5.9 Interrupt Interface AC characteristics

($Tc = 0 – 70°C$, $V_{CCIO} = 3.3\ V \pm 0.2\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Item	Symbol	Conditions	Min.	Max.	Unit
INT Input Pulse Width Time	t_{PW_INT}	Boot configuration ADDR[2]=H	$2 \times t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$1/2 \times t_{MCP} \times 1.1$	—	ns
NMI Input Pulse Width Time	t_{PW_NMI}	Boot configuration ADDR[2]=H	$t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$1/4 \times t_{MCP} \times 1.1$	—	ns

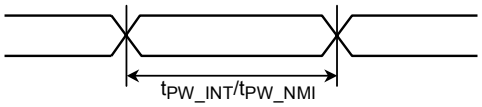


Figure 22.5.10 Timing Diagram: INT/NMI Interface

22.5.10 SIO Interface AC characteristics

($Tc = 0 – 70°C$, $V_{CCIO} = 3.3\ V \pm 0.2\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Item	Symbol	Conditions	Min.	Max.	Unit
SCLK Cycle time	f_{CYC_SCLK}	Boot configuration ADDR[2]=H	$4 \times t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$t_{MCP} \times 1.1$	—	ns
SCLK Frequency	f_{SCLK}	Boot configuration ADDR[2]=H	—	$1/2 \times f_{MCK} \times 0.45$	MHz
		Boot configuration ADDR[2]=L	—	$2 \times f_{MCK} \times 0.45$	MHz
SCLK High Time	t_{HIGH_SCLK}	Boot configuration ADDR[2]=H	$2 \times t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$1/2 \times t_{MCP} \times 1.1$	—	ns
SCLK Low Time	t_{LOW_SCLK}	Boot configuration ADDR[2]=H	$2 \times t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$1/2 \times t_{MCP} \times 1.1$	—	ns

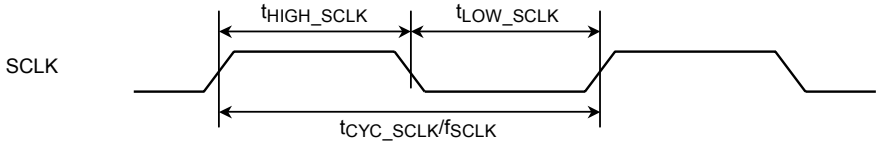


Figure 22.5.11 Timing Diagram: SIO Interface

22.5.11 Timer Interface AC characteristics

($Tc = 0 – 70°C$, $V_{CCIO} = 3.3\ V \pm 0.2\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Item	Symbol	Conditions	Min.	Max.	Unit
TCLK Cycle Time	f_{CYC_TCLK}	Boot configuration ADDR[2]=H	$4 \times t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$t_{MCP} \times 1.1$	—	ns
TCLK Frequency	f_{TCLK}	Boot configuration ADDR[2]=H	—	$1/2 \times f_{MCK} \times 0.45$	MHz
		Boot configuration ADDR[2]=L	—	$2 \times f_{MCK} \times 0.45$	MHz
TCLK High Time	t_{HIGH_TCLK}	Boot configuration ADDR[2]=H	$2 \times t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$1/2 \times t_{MCP} \times 1.1$	—	ns
TCLK Low Time	t_{LOW_TCLK}	Boot Configuration ADDR[2]=H	$2 \times t_{MCP} \times 1.1$	—	ns
		Boot configuration ADDR[2]=L	$1/2 \times t_{MCP} \times 1.1$	—	ns

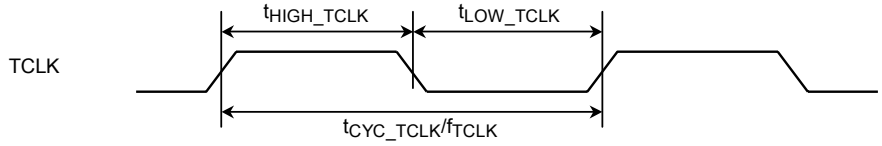


Figure 22.5.12 Timing Diagram: Timer Interface

22.5.12 PIO Interface AC characteristics

(Tc = 0 – 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
PIO[15:0] Output Delay Time	t_{VAL_PIO}	IMBUSCLK reference (C_L=50 pF)	—	10	ns
PIO[15:0] Input Setup Time	t_{SU_PIO}	IMBUSCLK reference	12	—	ns
PIO[15:0] Input Hold Time	t_{HO_PIO}	IMBUSCLK Reference	0	—	ns

*1) IMBUSCLK is an internal signal. For more details, see Chapter 6 Clocks.

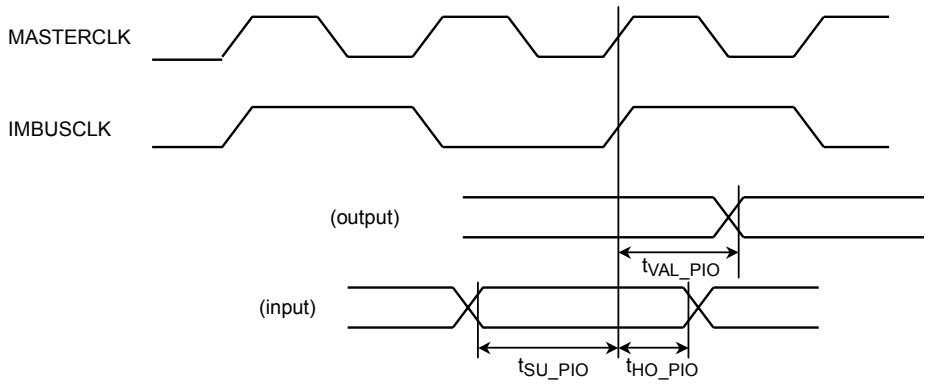


Figure 22.5.13 Timing Diagram: PIO Interface

22.5.13 AC-link Interface AC characteristics

(Tc = 0 – 70°C, V_{CCIO} = 3.3V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Condition	Min.	Max.	Unit
BITCLK High Time	t_{HIGH_BCLK}		36	45	ns
BITCLK Low Time	t_{LOW_BCLK}		36	45	ns
SYNC Output Delay Time	t_{VAL_SYNC}	BITCLK reference, CL = 55 pF	—	15	ns
SDOUT Output Delay Time	t_{VAL_SDOUT}	BITCLK reference, CL = 55 pF	—	15	ns
SDIN[1:0] Input Setup Time	t_{SU_DSIN}	BITCLK reference	10	—	ns
SDIN[1:0] Input Hold Time	t_{HO_DSIN}	BITCLK reference	10	—	ns

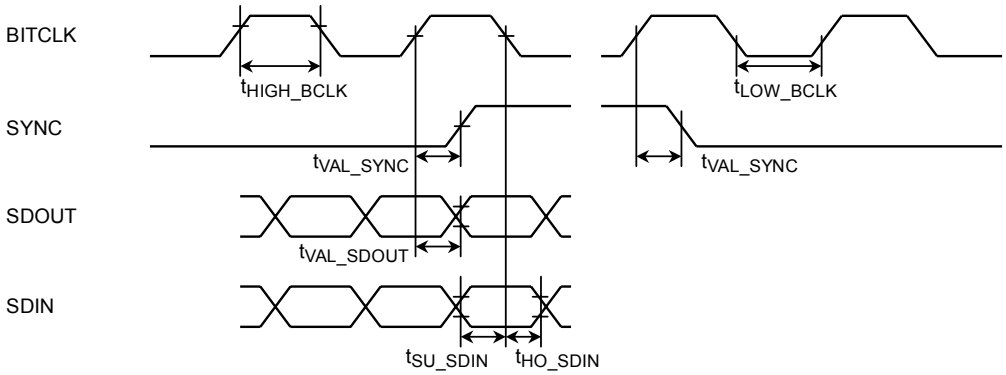


Figure 22.5.14 Timing Diagram: AC-link Interface

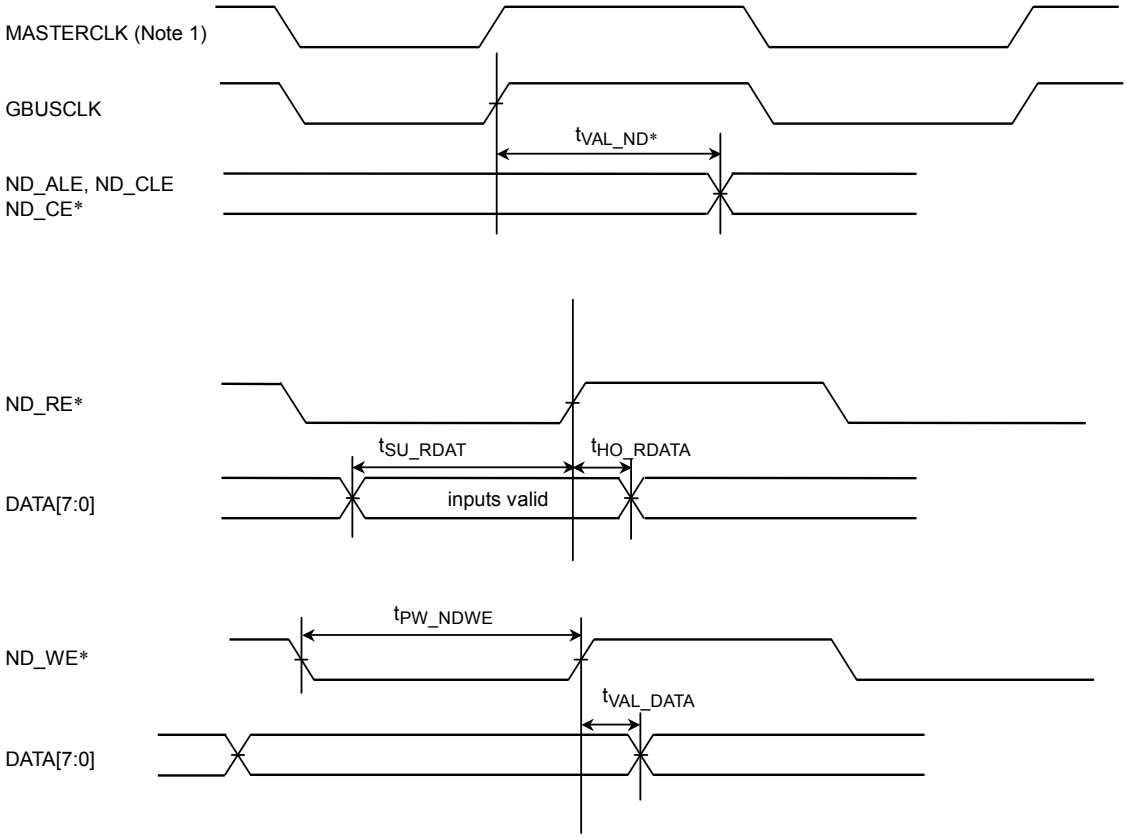
22.5.14 NAND Flash Memory Interface AC characteristics

(Tc = 0 – 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Condition	Min.	Max.	Unit
ND_ALE Output Delay Time	t_{VAL_NDALE}	GBUSCLK reference	—	15	ns
ND_CLE Output Delay Time	t_{VAL_NDCLE}	GBUSCLK reference	—	15	ns
ND_CE* Output Delay Time	t_{VAL_NDCE}	GBUSCLK reference	—	15	ns
DATA[7:0] Read Setup Time	t_{SU_RDATA}	ND_RE* rising edge	12	—	ns
DATA[7:0] Read Hold Time	t_{HO_RDATA}	ND_RE* rising edge	0	—	ns
ND_WE* Low Pulse Width	t_{PW_NDWE}	—	*1)	—	ns
DATA[7:0] Output Delay	t_{VAL_DATA}	ND_RE* rising edge	*2)	—	ns

*1) t_{VAL_NDWE} = (NDFSPR.SPW + 1) × GBUSCLK cycle

*2) t_{VAL_DATA} = NDFSPR.HOLD × GBUSCLK cycle – 3

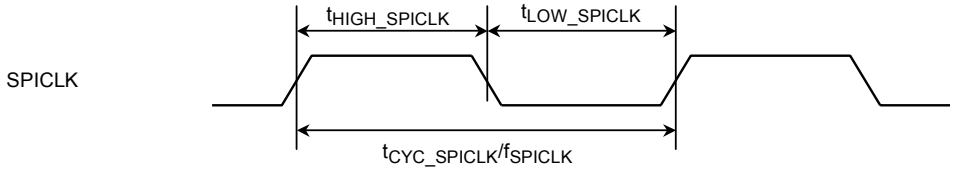


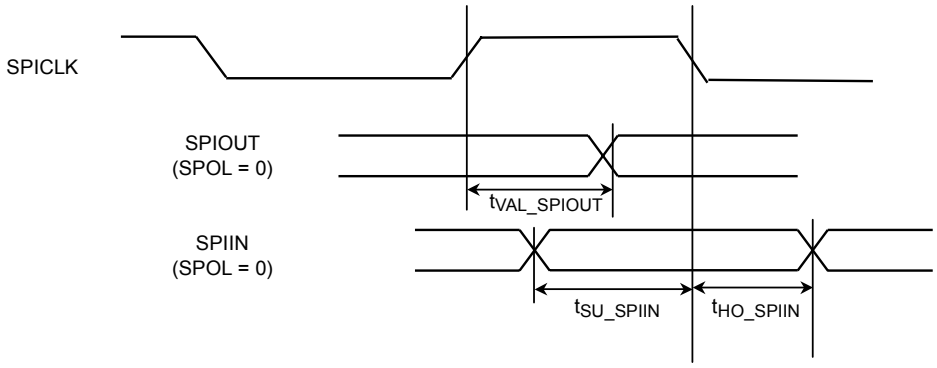
Note 1: This figure shows the timing diagram when the boot configuration setting is ADDR[2]=H. The NAND Flash Memory Controller operates synchronously to the internal GBUSCLK. GBUSCLK is an internal signal. When ADDR[2]=H, MASTERCLK and GBUSCLK become clocks with the same phase. When ADDR[2]=L, GBUSCLK has a frequency 4 times that of MASTERCLK. The timings of the rising edges match between both clocks.

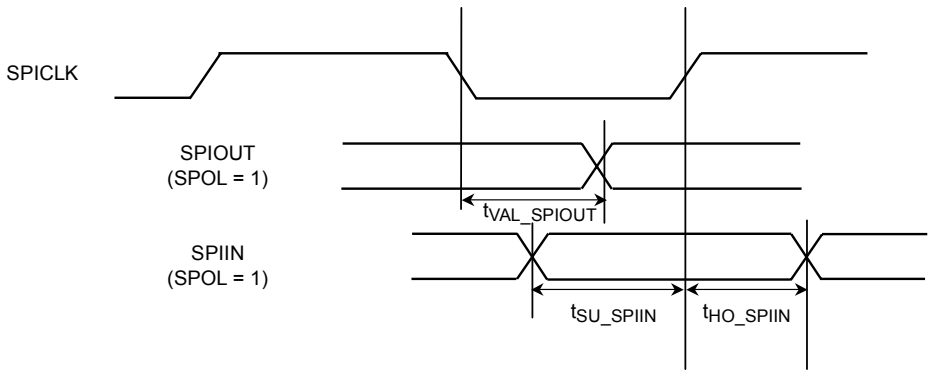
22.5.15 SPI AC characteristics

(Tc = 0 - 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
SPICLK Cycle Time	f_{CYC_SPICLK}	Boot Configuration ADDR[2]=H	$8 \times t_{MCP}$	—	ns
		Boot Configuration ADDR[2]=L	$2 \times t_{MCP}$	—	ns
SPICLK Frequency	f_{SPICLK}	Boot Configuration ADDR[2]=H	—	$1/8 \times f_{MCK}$	MHz
		Boot Configuration ADDR[2]=L	—	$1/2 \times f_{MCK}$	MHz
SPICLK High Time	t_{HIGH_SPICLK}		$t_{MCP} \times 0.9$	—	ns
SPICLK Low Time	t_{LOW_SPICLK}		$t_{MCP} \times 0.9$	—	ns
SPIOUT Output Delay Time	t_{VAL_SPIOUT}		—	6	ns
SPIIN Input Setup Time	t_{SU_SPIIN}		16	—	ns
SPIIN Input Hold Time	t_{HO_SPIIN}		0	—	ns





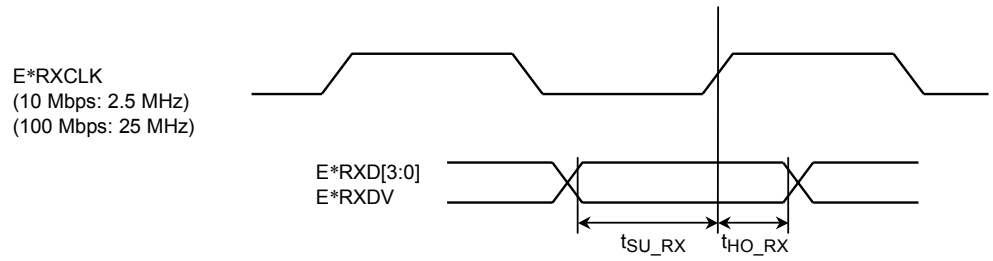


22.5.16 Ethernet Interface (MII) AC characteristics

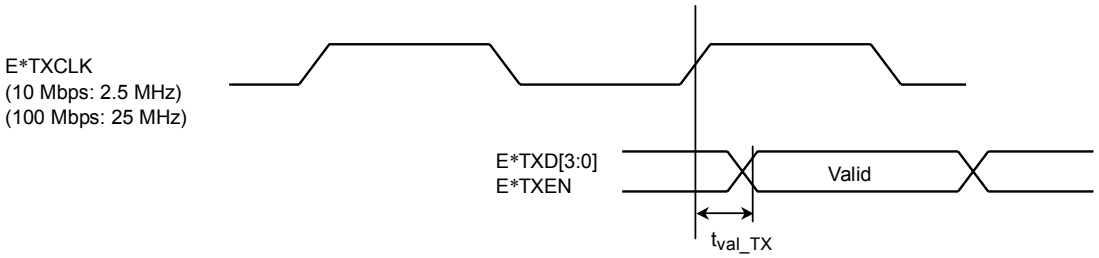
(Tc = 0 – 70°C, V_{CCIO} = 3.3 V ± 0.2 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
Reception Data Setup Time	t_{SU_RX}	E*RXCLK reference	10	—	ns
Reception Data Set Hold Time	t_{HO_RX}	E*RXCLK reference	10	—	ns
Transmission Data Output Delay Time	t_{val_TX}	E*TXCLK reference CLK = 50 pF	0	25	ns
E*MDC Cycle Time	t_{CYC_MDC}	When on-chip PCI bus is clocked at 1/4 of GBUS clock, (GBUSCLK=120 MHz, 120÷4÷14=2.14 MHz)	467	—	ns
		When on-chip PCI bus is clocked at 1/2 of GBUS clock, (GBUSCLK=120 MHz, 120÷2÷30=2 MHz)	500	—	ns
E*MDIO Input Setup Time	t_{SU_MDIO}	E*MDC reference	40	—	ns
E*MDIO Input Hold Time	t_{HO_MDIO}	E*MDC reference	0	—	ns
E*MDI Output Delay Time (PHY Setup)	$t_{VAL\_SU\_MDIO}$	E*MDC reference E*MDC CL = 30 pF E*MDIO CL = 50 pF	10	—	ns
E*MDI Output Delay Time (PHY Hold)	$t_{VAL\_HO\_MDIO}$		10	—	ns

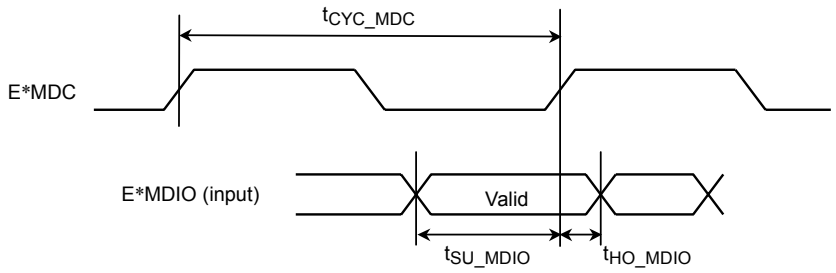
(1) Reception signals



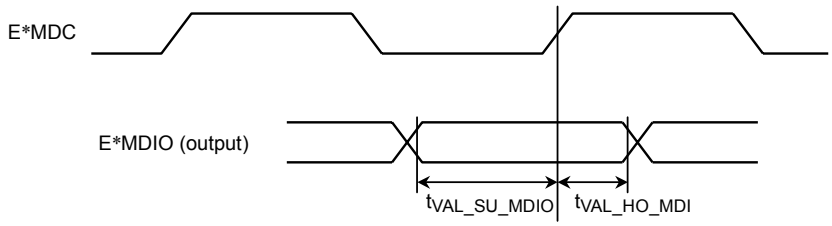
(2) Transmission signals



(3) Management signals



When TX4938 receives E*MDIO



When TX4938 outputs E*MDIO

23. Pinout and Package Information

23.1 Pinout Diagram

Figure 23.1.1 shows the TX4938 pinout. Table 23.1.1 provides a pin cross reference by pin number. provides a pin cross reference by pin name. Table 23.1.3 provides a pin cross reference for thermal balls.

	A	B	C	D	E	F	G	H	J	K	L	M	N
26	C_BE[2]	VSS	PCIAD [16]	PCIAD [18]	PCIAD [19]	PCIAD [23]	PCIAD [24]	PCICLK [0]	PCICLK [1]	PCICLK [2]	PCICLK [3]	PCICLK [4]	PCICLK [5]
25	IRDY*	FRAME*	PCIAD [17]	VddIO	PCIAD [20]	VddIO	PCIAD [25]	VddIO	GNT[0]*	REQ[0]*	GNT[2]*	GNT[3]*	DATA[63]
24	VddIO	STOP*	DEVSEL*	TRDY*	PCIAD [21]	ID_SEL	PCIAD [26]	PCIAD [29]	PCIAD [30]	GNT[1]*	REQ[2]*	REQ[3]*	VSS
23	VSS	PERR*	LOCK*	VddIN	PCIAD [22]	C_BE[3]	PCIAD [27]	VddIN	VSS	VddIN	VSS	VddIO	VddIO
22	PCIAD [15]	C_BE[1]	PAR	SERR*	VSS	VddIO	PCIAD [28]	VSS	PCIAD [31]	VSS	REQ[1]*	VSS	PME*
21	PCIAD [11]	PCIAD [12]	PCIAD [13]	VddIO	PCIAD [14]								
20	C_BE[0]	PCIAD[8]	PCIAD[9]	M66EN	PCIAD [10]								
19	PCIAD[5]	PCIAD[6]	PCIAD[7]	VddIO	VSS								
18	PCIAD[2]	PCIAD[3]	VddIO	PCIAD[4]	VSS								
17	PCST[0]	PCIAD[0]	PCIAD[1]	VddIN	VSS					VSS	VSS	VSS	VSS
16	PCST[3]	PCST[2]	PCST[1]	VddIO	TRST*					VSS	VSS	VSS	VSS
15	EEPROM_CS	PCST[5]	PCST[4]	VddIN	VSS					VSS	VSS	VSS	VSS
14	EEPROM_SK	PCST[8]	PCST[7]	VddIO	PCST[6]					VSS	VSS	VSS	VSS
13	VSS	TDO	TPC[3]	TPC[2]	TPC[1]					VSS	VSS	VSS	VSS
12	EEPROM_DO	DCLK	TMS	VddIO	VSS					VSS	VSS	VSS	VSS
11	EEPROM_DI	TCK	TDI	DMAACK [0]	DMAREQ [0]					VSS	VSS	VSS	VSS
10	BWE[1]*	BWE[2]*	BWE[3]*	VddIN	VSS					VSS	VSS	VSS	VSS
9	BWE[0]*	DMAREQ [1]	VddIO	VSS	DMADONE*								
8	DMAACK [1]	DMAREQ [2]	DMAREQ [3]	VddIN	VSS								
7	DMAACK [2]	DMAACK [3]	CE[1]*	CE[0]*	VddIO								
6	CE[4]*	CE[3]*	CE[2]*	VSS	SDIN[1]	TOP View							
5	CE[5]*	VddIO	ACE*	BYPASSPLL*	VSS	VddIN	VddIN	VSS	RXD[1]	VSS	TEST[1]*	VSS	VddIO
4	CE[7]*	CE[6]*	ACK*	VddIN	VddIO	NMI*	RXD[0]	VddIO	CTS[1]*	VddIN	TEST[2]*	VddIN	VSS
3	SWE*	BUSSPRT*	VddIO	PIO[6]	TIMER[1]	INT[0]	INT[3]	CTS[0]*	RTS[1]*	HALTDOZE	TEST[3]*	VddIO	DATA[0]
2	PIO[0]	PIO[2]	PIO[4]	VSS	TIMER[0]	INT[1]	INT[4]	RTS[0]*	TXD[1]	TEST[0]*	TEST[4]*	WDRST*	DATA[32]
1	PIO[1]	PIO[3]	PIO[5]	PIO[7]	TCLK	INT[2]	INT[5]	TXD[0]	SCLK	RESET*	SYSCLK	OE*	DATA[1]

Figure 23.1.1 Pinout Diagram (1/2)

P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	
PCICLKIN	MASTERCLK	VSS	VSS	DATA[29]	DATA[59]	DATA[58]	DATA[56]	VSS	DATA[22]	DATA[21]	VddIO	VSS	26
PLL2VSS_A	PLL1VSS_A	DATA[31]	DATA[61]	DATA[60]	DATA[27]	DATA[26]	DATA[24]	DATA[54]	DATA[53]	DATA[52]	DATA[20]	DATA[51]	25
PLL2VDD_A	PLL1VDD_A	VddIO	VddIO	VSS	VSS	DATA[57]	DATA[55]	VSS	VSS	DATA[50]	VddIO	DATA[19]	24
CGRESET*	VddIN	DATA[62]	VddIN	DATA[28]	VddIO	DATA[25]	VSS	VddIO	VddIN	VSS	VSS	VddIO	23
VddIN	VSS	DATA[30]	VSS	VddIO	VSS	VddIO	DATA[23]	VSS	DATA[48]	DATA[17]	DATA[49]	DATA[18]	22
								CB[3]	VSS	CB[7]	VddIO	DATA[16]	21
								DQM[7]	CB[2]	VSS	CB[6]	VddIO	20
								VSS	VddIN	VSS	DQM[3]	SDCLK[1]	19
								SDCS[3]*	DQM[2]	DQM[6]	VSS	VSS	18
VSS	VSS	VSS	VSS					VSS	VddIN	CKE	SDCS[2]*	SDCLK[3]	17
VSS	VSS	VSS	VSS					ADDR[17]	VddIO	ADDR[18]	ADDR[19]	VSS	16
VSS	VSS	VSS	VSS					VSS	VddIO	ADDR[15]	ADDR[16]	SDCLKIN	15
VSS	VSS	VSS	VSS					VSS	VddIO	ADDR[14]	VSS	SDCLK[0]	14
VSS	VSS	VSS	VSS					VSS	VddIO	ADDR[12]	ADDR[13]	SDCLK[2]	13
VSS	VSS	VSS	VSS					VSS	VddIN	ADDR[10]	VSS	ADDR[11]	12
VSS	VSS	VSS	VSS					ADDR[7]	ADDR[8]	VSS	ADDR[9]	VddIO	11
VSS	VSS	VSS	VSS					VSS	VddIN	ADDR[5]	ADDR[6]	VSS	10
								ADDR[3]	VddIO	VSS	ADDR[4]	VddIO	9
								VSS	VddIO	VSS	ADDR[1]	ADDR[2]	8
								VddIO	SDCS[0]*	SDCS[1]*	RAS*	ADDR[0]	7
							TOP View	DQM[0]	VSS	DQM[4]	DQM[1]	DQM[5]	6
VddIO	VSS	DATA[4]	VSS	DATA[7]	VSS	DATA[40]	DATA[42]	VSS	VddIO	VSS	WE*	VSS	5
VSS	VddIO	VddIO	VddIN	VddIO	VddIN	DATA[9]	VSS	DATA[12]	VddIN	VSS	CB[5]	CAS*	4
DATA[33]	DATA[34]	DATA[36]	DATA[37]	VddIO	VSS	VSS	VddIO	DATA[44]	VSS	CB[0]	CB[4]	CB[1]	3
VSS	DATA[3]	DATA[5]	DATA[6]	VSS	DATA[39]	DATA[41]	DATA[11]	DATA[13]	VSS	DATA[46]	VddIO	DATA[47]	2
DATA[2]	DATA[35]	VSS	DATA[38]	VSS	DATA[8]	DATA[10]	DATA[43]	DATA[45]	DATA[14]	VddIO	DATA[15]	VSS	1

Figure 23.1.1 Pinout Diagram (2/2)

Table 23.1.1 Pin Cross Reference by Pin Number (1/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	PIO[1]	B17	PCIAD[0]	D7	CE[0]*	E23	PCIAD[22]	J25	GNT[0]*
A2	PIO[0]	B18	PCIAD[3]	D8	VddIN	E24	PCIAD[21]	J26	PCICLK[1]
A3	SWE*	B19	PCIAD[6]	D9	VSS	E25	PCIAD[20]	K1	RESET*
A4	CE[7]*	B20	PCIAD[8]	D10	VddIN	E26	PCIAD[19]	K2	TEST[0]*
A5	CE[5]*	B21	PCIAD[12]	D11	DMAACK[0]	F1	INT[2]	K3	HALTDOZE
A6	CE[4]*	B22	C_BE[1]	D12	VddIO	F2	INT[1]	K4	VddIN
A7	DMAACK[2]	B23	PERR*	D13	TPC[2]	F3	INT[0]	K5	VSS
A8	DMAACK[1]	B24	STOP*	D14	VddIO	F4	NMI*	K22	VSS
A9	BWE[0]*	B25	FRAME*	D15	VddIN	F5	VddIN	K23	VddIN
A10	BWE[1]*	B26	VSS	D16	VddIO	F22	VddIO	K24	GNT[1]*
A11	EEPROM_DI	C1	PIO[5]	D17	VddIN	F23	C_BE[3]	K25	REQ[0]*
A12	EEPROM_DO	C2	PIO[4]	D18	PCIAD[4]	F24	ID_SEL	K26	PCICLK[2]
A13	VSS	C3	VddIO	D19	VddIO	F25	VddIO	L1	SYSCLK
A14	EEPROM_SK	C4	ACK*	D20	M66EN	F26	PCIAD[23]	L2	TEST[4]*
A15	EEPROM_CS	C5	ACE*	D21	VddIO	G1	INT[5]	L3	TEST[3]*
A16	PCST[3]	C6	CE[2]*	D22	SERR*	G2	INT[4]	L4	TEST[2]*
A17	PCST[0]	C7	CE[1]*	D23	VddIN	G3	INT[3]	L5	TEST[1]*
A18	PCIAD[2]	C8	DMAREQ[3]	D24	TRDY*	G4	RXD[0]	L22	REQ[1]*
A19	PCIAD[5]	C9	VddIO	D25	VddIO	G5	VddIN	L23	VSS
A20	C_BE[0]	C10	BWE[3]*	D26	PCIAD[18]	G22	PCIAD[28]	L24	REQ[2]*
A21	PCIAD[11]	C11	TDI	E1	TCLK	G23	PCIAD[27]	L25	GNT[2]*
A22	PCIAD[15]	C12	TMS	E2	TIMER[0]	G24	PCIAD[26]	L26	PCICLK[3]
A23	VSS	C13	TPC[3]	E3	TIMER[1]	G25	PCIAD[25]	M1	OE*
A24	VddIO	C14	PCST[7]	E4	VddIO	G26	PCIAD[24]	M2	WDRST*
A25	IRDY*	C15	PCST[4]	E5	VSS	H1	TXD[0]	M3	VddIO
A26	C_BE[2]	C16	PCST[1]	E6	SDIN[1]	H2	RTS[0]*	M4	VddIN
B1	PIO[3]	C17	PCIAD[1]	E7	VddIO	H3	CTS[0]*	M5	VSS
B2	PIO[2]	C18	VddIO	E8	VSS	H4	VddIO	M22	VSS
B3	BUSSPRT*	C19	PCIAD[7]	E9	DMADONE*	H5	VSS	M23	VddIO
B4	CE[6]*	C20	PCIAD[9]	E10	VSS	H22	VSS	M24	REQ[3]*
B5	VddIO	C21	PCIAD[13]	E11	DMAREQ[0]	H23	VddIN	M25	GNT[3]*
B6	CE[3]*	C22	PAR	E12	VSS	H24	PCIAD[29]	M26	PCICLK[4]
B7	DMAACK[3]	C23	LOCK*	E13	TPC[1]	H25	VddIO	N1	DATA[1]
B8	DMAREQ[2]	C24	DEVSEL*	E14	PCST[6]	H26	PCICLK[0]	N2	DATA[32]
B9	DMAREQ[1]	C25	PCIAD[17]	E15	VSS	J1	SCLK	N3	DATA[0]
B10	BWE[2]*	C26	PCIAD[16]	E16	TRST*	J2	TXD[1]	N4	VSS
B11	TCK	D1	PIO[7]	E17	VSS	J3	RTS[1]*	N5	VddIO
B12	DCLK	D2	VSS	E18	VSS	J4	CTS[1]*	N22	PME*
B13	TDO	D3	PIO[6]	E19	VSS	J5	RXD[1]	N23	VddIO
B14	PCST[8]	D4	VddIN	E20	PCIAD[10]	J22	PCIAD[31]	N24	VSS
B15	PCST[5]	D5	BYPASSPLL*	E21	PCIAD[14]	J23	VSS	N25	DATA[63]
B16	PCST[2]	D6	VSS	E22	VSS	J24	PCIAD[30]	N26	PCICLK[5]

Table 23.1.1 Pin Cross Reference by Pin Number (2/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
P1	DATA[2]	V3	VddIO	AB5	VSS	AC21	VSS	AE11	ADDR[9]
P2	VSS	V4	VddIO	AB6	DQM[0]	AC22	DATA[48]	AE12	VSS
P3	DATA[33]	V5	DATA[7]	AB7	VddIO	AC23	VddIN	AE13	ADDR[13]
P4	VSS	V22	VddIO	AB8	VSS	AC24	VSS	AE14	VSS
P5	VddIO	V23	DATA[28]	AB9	ADDR[3]	AC25	DATA[53]	AE15	ADDR[16]
P22	VddIN	V24	VSS	AB10	VSS	AC26	DATA[22]	AE16	ADDR[19]
P23	CGRESET*	V25	DATA[60]	AB11	ADDR[7]	AD1	VddIO	AE17	SDCS[2]*
P24	PLL2VDD_A	V26	DATA[29]	AB12	VSS	AD2	DATA[46]	AE18	VSS
P25	PLL2VSS_A	W1	DATA[8]	AB13	VSS	AD3	CB[0]	AE19	DQM[3]
P26	PCICLKIN	W2	DATA[39]	AB14	VSS	AD4	VSS	AE20	CB[6]
R1	DATA[35]	W3	VSS	AB15	VSS	AD5	VSS	AE21	VddIO
R2	DATA[3]	W4	VddIN	AB16	ADDR[17]	AD6	DQM[4]	AE22	DATA[49]
R3	DATA[34]	W5	VSS	AB17	VSS	AD7	SDCS[1]*	AE23	VSS
R4	VddIO	W22	VSS	AB18	SDCS[3]*	AD8	VSS	AE24	VddIO
R5	VSS	W23	VddIO	AB19	VSS	AD9	VSS	AE25	DATA[20]
R22	VSS	W24	VSS	AB20	DQM[7]	AD10	ADDR[5]	AE26	VddIO
R23	VddIN	W25	DATA[27]	AB21	CB[3]	AD11	VSS	AF1	VSS
R24	PLL1VDD_A	W26	DATA[59]	AB22	VSS	AD12	ADDR[10]	AF2	DATA[47]
R25	PLL1VSS_A	Y1	DATA[10]	AB23	VddIO	AD13	ADDR[12]	AF3	CB[1]
R26	MASTERCLK	Y2	DATA[41]	AB24	VSS	AD14	ADDR[14]	AF4	CAS*
T1	VSS	Y3	VSS	AB25	DATA[54]	AD15	ADDR[15]	AF5	VSS
T2	DATA[5]	Y4	DATA[9]	AB26	VSS	AD16	ADDR[18]	AF6	DQM[5]
T3	DATA[36]	Y5	DATA[40]	AC1	DATA[14]	AD17	CKE	AF7	ADDR[0]
T4	VddIO	Y22	VddIO	AC2	VSS	AD18	DQM[6]	AF8	ADDR[2]
T5	DATA[4]	Y23	DATA[25]	AC3	VSS	AD19	VSS	AF9	VddIO
T22	DATA[30]	Y24	DATA[57]	AC4	VddIN	AD20	VSS	AF10	VSS
T23	DATA[62]	Y25	DATA[26]	AC5	VddIO	AD21	CB[7]	AF11	VddIO
T24	VddIO	Y26	DATA[58]	AC6	VSS	AD22	DATA[17]	AF12	ADDR[11]
T25	DATA[31]	AA1	DATA[43]	AC7	SDCS[0]*	AD23	VSS	AF13	SDCLK[2]
T26	VSS	AA2	DATA[11]	AC8	VddIO	AD24	DATA[50]	AF14	SDCLK[0]
U1	DATA[38]	AA3	VddIO	AC9	VddIO	AD25	DATA[52]	AF15	SDCLKIN
U2	DATA[6]	AA4	VSS	AC10	VddIN	AD26	DATA[21]	AF16	VSS
U3	DATA[37]	AA5	DATA[42]	AC11	ADDR[8]	AE1	DATA[15]	AF17	SDCLK[3]
U4	VddIN	AA22	DATA[23]	AC12	VddIN	AE2	VddIO	AF18	VSS
U5	VSS	AA23	VSS	AC13	VddIO	AE3	CB[4]	AF19	SDCLK[1]
U22	VSS	AA24	DATA[55]	AC14	VddIO	AE4	CB[5]	AF20	VddIO
U23	VddIN	AA25	DATA[24]	AC15	VddIO	AE5	WE*	AF21	DATA[16]
U24	VddIO	AA26	DATA[56]	AC16	VddIO	AE6	DQM[1]	AF22	DATA[18]
U25	DATA[61]	AB1	DATA[45]	AC17	VddIN	AE7	RAS*	AF23	VddIO
U26	VSS	AB2	DATA[13]	AC18	DQM[2]	AE8	ADDR[1]	AF24	DATA[19]
V1	VSS	AB3	DATA[44]	AC19	VddIN	AE9	ADDR[4]	AF25	DATA[51]
V2	VSS	AB4	DATA[12]	AC20	CB[2]	AE10	ADDR[6]	AF26	VSS

Table 23.1.2 Pin Cross Reference by Pin Name (1/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
C5	ACE*	A5	CE[5]*	T22	DATA[30]	B9	DMAREQ[1]	C19	PCIAD[7]
C4	ACK*	B4	CE[6]*	T25	DATA[31]	B8	DMAREQ[2]	B20	PCIAD[8]
AF7	ADDR[0]	A4	CE[7]*	N2	DATA[32]	C8	DMAREQ[3]	C20	PCIAD[9]
AE8	ADDR[1]	P23	CGRESET*	P3	DATA[33]	AB6	DQM[0]	E20	PCIAD[10]
AF8	ADDR[2]	AD17	CKE	R3	DATA[34]	AE6	DQM[1]	A21	PCIAD[11]
AB9	ADDR[3]	K1	RESET*	R1	DATA[35]	AC18	DQM[2]	B21	PCIAD[12]
AE9	ADDR[4]	H3	CTS[0]*	T3	DATA[36]	AE19	DQM[3]	C21	PCIAD[13]
AD10	ADDR[5]	J4	CTS[1]*	U3	DATA[37]	AD6	DQM[4]	E21	PCIAD[14]
AE10	ADDR[6]	A20	C_BE[0]	U1	DATA[38]	AF6	DQM[5]	A22	PCIAD[15]
AB11	ADDR[7]	B22	C_BE[1]	W2	DATA[39]	AD18	DQM[6]	C26	PCIAD[16]
AC11	ADDR[8]	A26	C_BE[2]	Y5	DATA[40]	AB20	DQM[7]	C25	PCIAD[17]
AE11	ADDR[9]	F23	C_BE[3]	Y2	DATA[41]	A15	EEPROM_CS	D26	PCIAD[18]
AD12	ADDR[10]	N3	DATA[0]	AA5	DATA[42]	A11	EEPROM_DI	E26	PCIAD[19]
AF12	ADDR[11]	N1	DATA[1]	AA1	DATA[43]	A12	EEPROM_DO	E25	PCIAD[20]
AD13	ADDR[12]	P1	DATA[2]	AB3	DATA[44]	A14	EEPROM_SK	E24	PCIAD[21]
AE13	ADDR[13]	R2	DATA[3]	AB1	DATA[45]	B25	FRAME*	E23	PCIAD[22]
AD14	ADDR[14]	T5	DATA[4]	AD2	DATA[46]	J25	GNT[0]*	F26	PCIAD[23]
AD15	ADDR[15]	T2	DATA[5]	AF2	DATA[47]	K24	GNT[1]*	G26	PCIAD[24]
AE15	ADDR[16]	U2	DATA[6]	AC22	DATA[48]	L25	GNT[2]*	G25	PCIAD[25]
AB16	ADDR[17]	V5	DATA[7]	AE22	DATA[49]	M25	GNT[3]*	G24	PCIAD[26]
AD16	ADDR[18]	W1	DATA[8]	AD24	DATA[50]	K3	HALTDOZE	G23	PCIAD[27]
AE16	ADDR[19]	Y4	DATA[9]	AF25	DATA[51]	F24	ID_SEL	G22	PCIAD[28]
B3	BUSSPRT*	Y1	DATA[10]	AD25	DATA[52]	F3	INT[0]	H24	PCIAD[29]
A9	BWE[0]*	AA2	DATA[11]	AC25	DATA[53]	F2	INT[1]	J24	PCIAD[30]
A10	BWE[1]*	AB4	DATA[12]	AB25	DATA[54]	F1	INT[2]	J22	PCIAD[31]
B10	BWE[2]*	AB2	DATA[13]	AA24	DATA[55]	G3	INT[3]	P26	PCICLKIN
C10	BWE[3]*	AC1	DATA[14]	AA26	DATA[56]	G2	INT[4]	H26	PCICLK[0]
D5	BYPASSPLL*	AE1	DATA[15]	Y24	DATA[57]	G1	INT[5]	J26	PCICLK[1]
AF4	CAS*	AF21	DATA[16]	Y26	DATA[58]	A25	IRDY*	K26	PCICLK[2]
AD3	CB[0]	AD22	DATA[17]	W26	DATA[59]	C23	LOCK*	L26	PCICLK[3]
AF3	CB[1]	AF22	DATA[18]	V25	DATA[60]	D20	M66EN	M26	PCICLK[4]
AC20	CB[2]	AF24	DATA[19]	U25	DATA[61]	R26	MASTERCLK	N26	PCICLK[5]
AB21	CB[3]	AE25	DATA[20]	T23	DATA[62]	F4	NMI*	A17	PCST[0]
AE3	CB[4]	AD26	DATA[21]	N25	DATA[63]	M1	OE*	C16	PCST[1]
AE4	CB[5]	AC26	DATA[22]	B12	DCLK	C22	PAR	B16	PCST[2]
AE20	CB[6]	AA22	DATA[23]	C24	DEVSEL*	B17	PCIAD[0]	A16	PCST[3]
AD21	CB[7]	AA25	DATA[24]	D11	DMAACK[0]	C17	PCIAD[1]	C15	PCST[4]
D7	CE[0]*	Y23	DATA[25]	A8	DMAACK[1]	A18	PCIAD[2]	B15	PCST[5]
C7	CE[1]*	Y25	DATA[26]	A7	DMAACK[2]	B18	PCIAD[3]	E14	PCST[6]
C6	CE[2]*	W25	DATA[27]	B7	DMAACK[3]	D18	PCIAD[4]	C14	PCST[7]
B6	CE[3]*	V23	DATA[28]	E9	DMADONE*	A19	PCIAD[5]	B14	PCST[8]
A6	CE[4]*	V26	DATA[29]	E11	DMAREQ[0]	B19	PCIAD[6]	B23	PERR*

Table 23.1.2 Pin Cross Reference by Pin Name (2/2)

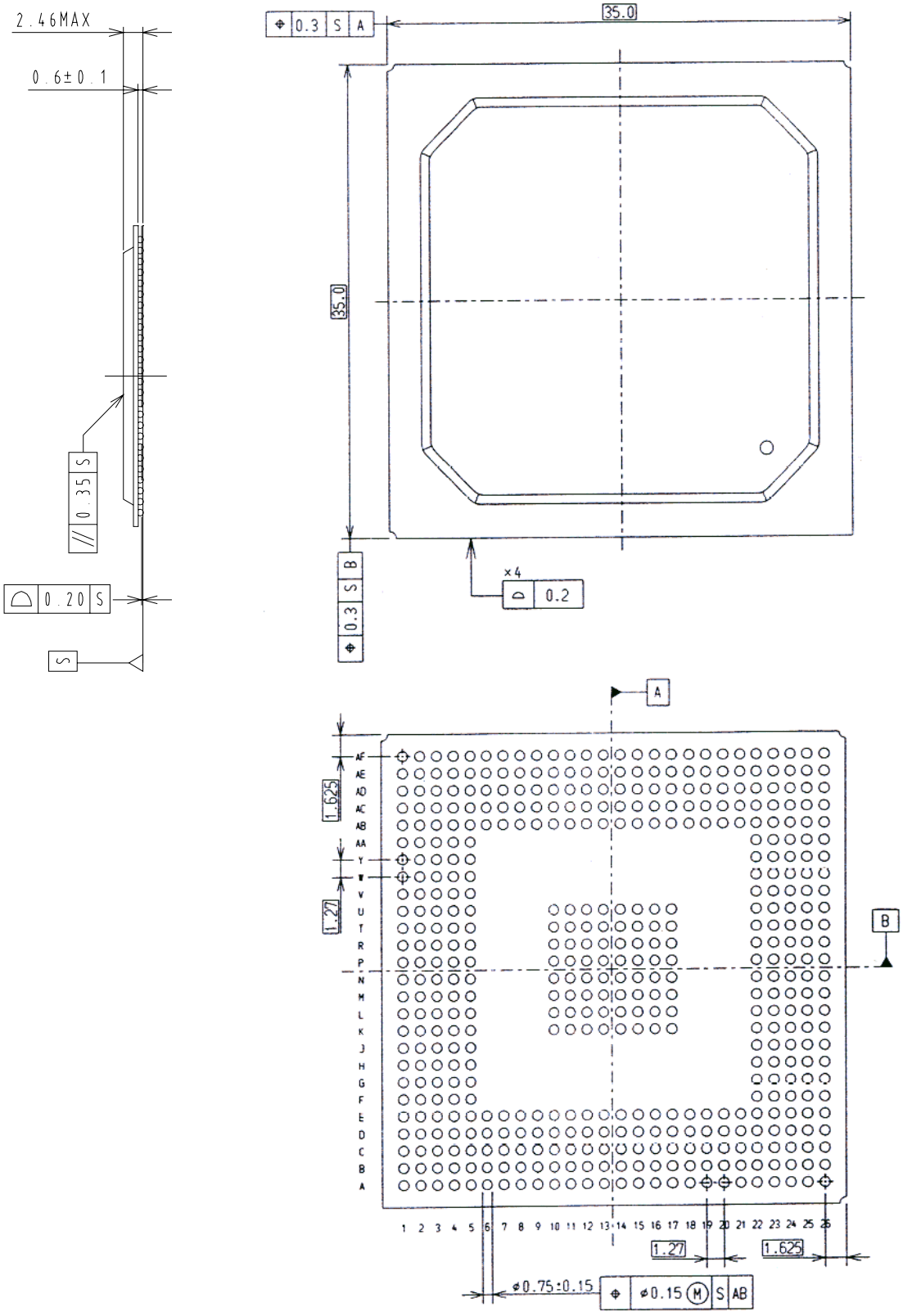
Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A2	PIO[0]	L5	TEST[1]*	R5	VSS	AE14	VSS	D25	VddIO
A1	PIO[1]	L4	TEST[2]*	T1	VSS	AE18	VSS	E4	VddIO
B2	PIO[2]	L3	TEST[3]*	T26	VSS	AE23	VSS	E7	VddIO
B1	PIO[3]	L2	TEST[4]*	U22	VSS	AF1	VSS	F22	VddIO
C2	PIO[4]	E2	TIMER[0]	U26	VSS	AF10	VSS	F25	VddIO
C1	PIO[5]	E3	TIMER[1]	U5	VSS	AF16	VSS	H25	VddIO
D3	PIO[6]	C12	TMS	V1	VSS	AF18	VSS	H4	VddIO
D1	PIO[7]	E13	TPC[1]	V2	VSS	AF26	VSS	M23	VddIO
R24	PLL1VDD_A	D13	TPC[2]	V24	VSS	AF5	VSS	M3	VddIO
R25	PLL1VSS_A	C13	TPC[3]	W22	VSS	D10	VddIN	N23	VddIO
P24	PLL2VDD_A	D24	TRDY*	W24	VSS	D15	VddIN	N5	VddIO
P25	PLL2VSS_A	E16	TRST*	W3	VSS	D17	VddIN	P5	VddIO
N22	PME*	H1	TXD[0]	W5	VSS	D23	VddIN	R4	VddIO
AE7	RAS*	J2	TXD[1]	Y3	VSS	D4	VddIN	T24	VddIO
K25	REQ[0]*	A13	VSS	AA23	VSS	D8	VddIN	T4	VddIO
L22	REQ[1]*	A23	VSS	AA4	VSS	F5	VddIN	U24	VddIO
L24	REQ[2]*	B26	VSS	AB10	VSS	G5	VddIN	V22	VddIO
M24	REQ[3]*	D2	VSS	AB12	VSS	H23	VddIN	V3	VddIO
H2	RTS[0]*	D6	VSS	AB13	VSS	K23	VddIN	V4	VddIO
J3	RTS[1]*	D9	VSS	AB14	VSS	K4	VddIN	W23	VddIO
G4	RXD[0]	E10	VSS	AB15	VSS	M4	VddIN	Y22	VddIO
J5	RXD[1]	E12	VSS	AB17	VSS	P22	VddIN	AA3	VddIO
J1	SCLK	E15	VSS	AB19	VSS	R23	VddIN	AB23	VddIO
E6	SDIN[1]	E17	VSS	AB22	VSS	U23	VddIN	AB7	VddIO
AF15	SDCLKIN	E18	VSS	AB24	VSS	U4	VddIN	AC13	VddIO
AF14	SDCLK[0]	E19	VSS	AB26	VSS	W4	VddIN	AC14	VddIO
AF19	SDCLK[1]	E22	VSS	AB5	VSS	AC10	VddIN	AC15	VddIO
AF13	SDCLK[2]	E5	VSS	AB8	VSS	AC12	VddIN	AC16	VddIO
AF17	SDCLK[3]	E8	VSS	AC2	VSS	AC17	VddIN	AC5	VddIO
AC7	SDCS[0]*	H22	VSS	AC21	VSS	AC19	VddIN	AC8	VddIO
AD7	SDCS[1]*	H5	VSS	AC24	VSS	AC23	VddIN	AC9	VddIO
AE17	SDCS[2]*	J23	VSS	AC3	VSS	AC4	VddIN	AD1	VddIO
AB18	SDCS[3]*	K22	VSS	AC6	VSS	A24	VddIO	AE2	VddIO
D22	SERR*	K5	VSS	AD11	VSS	B5	VddIO	AE21	VddIO
B24	STOP*	L23	VSS	AD19	VSS	C18	VddIO	AE24	VddIO
A3	SWE*	M22	VSS	AD20	VSS	C3	VddIO	AE26	VddIO
L1	SYSCLK	M5	VSS	AD23	VSS	C9	VddIO	AF11	VddIO
B11	TCK	N24	VSS	AD4	VSS	D12	VddIO	AF20	VddIO
E1	TCLK	N4	VSS	AD5	VSS	D14	VddIO	AF23	VddIO
C11	TDI	P2	VSS	AD8	VSS	D16	VddIO	AF9	VddIO
B13	TDO	P4	VSS	AD9	VSS	D19	VddIO	M2	WDRST*
K2	TEST[0]*	R22	VSS	AE12	VSS	D21	VddIO	AE5	WE*

Table 23.1.3 Pin Cross Reference for Thermal Balls

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
K10	VSS	L15	VSS	N12	VSS	P17	VSS	T14	VSS
K11	VSS	L16	VSS	N13	VSS	R10	VSS	T15	VSS
K12	VSS	L17	VSS	N14	VSS	R11	VSS	T16	VSS
K13	VSS	M10	VSS	N15	VSS	R12	VSS	T17	VSS
K14	VSS	M11	VSS	N16	VSS	R13	VSS	U10	VSS
K15	VSS	M12	VSS	N17	VSS	R14	VSS	U11	VSS
K16	VSS	M13	VSS	P10	VSS	R15	VSS	U12	VSS
K17	VSS	M14	VSS	P11	VSS	R16	VSS	U13	VSS
L10	VSS	M15	VSS	P12	VSS	R17	VSS	U14	VSS
L11	VSS	M16	VSS	P13	VSS	T10	VSS	U15	VSS
L12	VSS	M17	VSS	P14	VSS	T11	VSS	U16	VSS
L13	VSS	N10	VSS	P15	VSS	T12	VSS	U17	VSS
L14	VSS	N11	VSS	P16	VSS	T13	VSS		

23.2 Package Dimensions

P-BGA484-3535-1.27B9 Unit: mm



24. Notes on Use of TMPR4938

24.1 Notes on TX49/H3 Core

- Restriction on detect of the Bus errors when a data cycle generated by load instruction.

[Restriction]

Error notification to the TX49/H3 Core using Bus errors is not enabled or Executing a SYNC instruction immediately after the preceding load instruction.

[Violation]

When a Bus error exception (DBE) occurs during a data Read cycle generated by a preceding load instruction and an exception with a higher priority than the Bus error exception (DBE) occurs in a subsequently executed instruction, the exception of the subsequent instruction is processed first and Bus error exceptions (DBE) are no longer detected.

<Conditions>

The TX49/H3 Core has a non-blocking load function. With this function, the instruction that follows the preceding load instruction is executed without stalling if it is not dependent on the preceding load instruction.

When reading the data from the preceding load instruction and a Bus Error exception (DBE) occurs and an exception (see the following table for the priority order when consecutive instructions issue multiple exceptions at the same timing) with a higher priority than the Bus Error exception (DBE) of the subsequently executed instruction occurs, the exception that the subsequent instruction issued is processed before the Bus Error exception (DBE) and Bus Error exceptions (DBE) can no longer be detected.

Priority Order for Exceptions Issued at the Same Timing

Priority Sequence (High)		Detected PipeStage	Instruction synchronous or asynchronous
Cold Reset		M	Async
Soft Reset		M	Async
NMI		M	Async
Bus Error (IBE)	Instruction Fetch	M	Async
Ov,Tr,Sys,Bp,RI,CpU,FPE		M	Sync
Address Error (AdEL/AdES)	Data Access	M	Sync
TLB Refill (TLBL/TLBS)	Data Access	M	Sync
TLB Invalid (TLBL/TLBS)	Data Access	M	Sync
TLB Modify (TLBL/TLBS)	Data Access	M	Sync
Bus Error (DBE)	Data Access	M	Async
Interrupt		M	Async
Address Error (AdEL)	Instruction Fetch	E	Sync
TLB Refill (TLBL)	Instruction Fetch	E	Sync
TLB Invalid (TLBL)	Instruction Fetch	E	Sync

Note: The table above differs from Table 11-3 (Priority Order when the Same Instruction Issues Multiple Exceptions at the Same Timing) on page 11-2 of the “TX49/H2, H3, H4 Core Architecture”.

Bus errors occur the following three conditions.

(1) When CCFG.TOE of the Chip Configuration Register is set to "1" (Default: 0), G-Bus timeout error detection is enabled, and the following situation results:
- A Bus timeout occurs when a G-Bus Bus Master (TX49/H3 Core, DMAC, or PCIC) is reading the G-Bus
- A Bus timeout occurs when a G-Bus Bus Master (other than the TX49/H3 Core) is writing to the G-Bus

(2) When ECCCR.MEB of the ECCCR Register in the SDRAM Controller is set to "1" (Default: 0), Parity errors are enabled during a multi-bit error, and the following situation results:
- A 2-bit ECC error or Parity error is detected during SDRAM Read operation by the TX49/H3 Core
- A 2-bit ECC error or Parity error is detected during Read/Write operation by a G-Bus Bus Master other than the TX49/H3 Core

(3) When PCICCFG.IRBER of the PCICCFG Register in the PCI Controller is set to "1" (Default: 1), and the following situation results during initiator Read operation:
- A Parity error is detected
- A Master ABORT is received
- A Target ABORT is received
- A TRDY timeout is detected
- A Retry timeout is detected

There is no problem for ColdReset or SoftReset exceptions because initialization processing is performed after the exception occurs. Also, there is no problem for NMI exceptions if the process after the exception occurs is similar to the above reset process.

[Workaround]
- There is no problem if error notification to the TX49/H3 Core using Bus errors is not enabled in the above Conditions.
- Executing a SYNC instruction immediately after the preceding load instruction allows you to avoid this condition because the next instruction will not be executed until the Load data arrives.

24.2 Notes on External Bus Controller

- Output delay of DATA[63:0] depends on the external bus speed set with EBCCRn.SP. For details on it, see 22.5.4 External Bus Interface AC Characteristics.

24.3 Notes on DMA Controller

- When burst transfer is performed during DMA transfer, the increment value setting for address is restricted. For details, see 8.3.7 Single Address Transfer and 8.3.8 Dual Address Transfer.
- Restrictions in dual address transfer by the DMA controller

[Restriction]

Setting the DMA master control register (DMMCRn) is restricted in the dual address transfer mode.

[Violation]

When a bus error occurs on a channel contained in DMAC, “all 0” may be written repeatedly to the address (DMSARn), that should be read, in all channels including it.

In this case, the address value (DMSARn) and count register value (DMCNTRn) are not changed, and write is continued to the same address until CPU terminates DMA transfer (0 is set to DMCCRn.XFACT).

<Conditions>

This violation occurs in the following conditions.

(1) FIFO is disabled. 0 is set to either one channel or more of FIFUM[n] (n=3 to 0) in DMMCR.

(2) The channel shown above (1) is set to dual address transfer. 0 is set to SNGAD in DMCCRn (n=3 to 0).

(3) A bus error occurs during access to the destination address in the channel which satisfies both (1) and (2). 1 is set to DESERR in DMCSRn (n=3 to 0).

(4) When (3) is satisfied, single transfer (non-burst transfer) in dual address transfer mode (see Supplemental Remarks for details) performed in any channel causes the violation shown above. When DMMCR.FIFUM[n]=0 is set in dual address transfer mode, single transfer (non-burst transfer) is performed regardless of the value set to DMCCRn.XFSZ.

In the case only one channel is used, restarting DMS transfer without a reset of FIFO results in a malfunction when the conditions (1) to (3) are satisfied.

When two channels are used in a system, the following table shows settings to cause malfunctions.

Ch. A of a bus error			Ch. B			Transfer restarted in ch. A after a bus error (without reset of FIFO)	Ch. B after a bus error
FIFO	XFSZ	Transfer	FIFO	XFSZ	Transfer		
Disable	Any value	Single	Disable	Any value	Single	Malfunction of ch. A	Malfunction of ch. B
Disable	Any value	Single	Enable	<4DW	Single	Malfunction of ch. A	Malfunction of ch. B
Disable	Any value	Single	Enable	≥4DW	Burst 1)	Malfunction of ch. A	Correct operation of ch. B 1)

1) By this setting, Channel B performs burst transfer. When the different off-set values are set to the source address and destination address of Channel B, or source and destination burst inhibit bits are set, single and burst transfer modes are combined for data transfer, but a malfunction does not occur.

<Supplemental information> Dual address transfer mode

– Single transfer: Disable FIFO (DMMCR.FIFUM[n]=0]
or the value smaller than 4DW is set to DMCCRn.XFSZ.

– Burst transfer: Enable FIFO (DMMCR.FIFUM[n]=1]
and 4DW or larger value is set to DMCCRn.XFSZ.

[Workarounds]

Workarounds vary depending on which type of dual address transfer is performed together with dual address single transfer. Combination of 4 channels in DMAC0 or DMAC1 affects a workaround to be required.(DMAC0 and DMAC1 are independent controllers.)

	Single Transfer	Burst Transfer	Workaround
(1)	1 ch.	Null	(a) or (b)
(2)	1 ch.	1 to 3 ch.	(b)
(3)	2 to 4 ch.	Null	(a)
(4)	2 to 3 ch.	1 to 2 ch.	(a)

(a) Enable FIFO for all channels that perform single transfer.
(DMMCRn.FIFUM[n]=1: n=3 to 0)

(b) After all channels finished transfers, set 1 to RSFIF in DMMCRn, then write 0 to reset and release reset of FIFO.

– Only one channel performs single transfer

(1) Single transfer performed with only one channel

Perform either (a) or (b) to prevent a malfunction.

(2) Single transfer performed with one channel and burst transfer

Perform (b) to prevent a malfunction.

– Two or more channels perform single transfer

(3) Single transfer (performed with two to four channels)

Perform (a) to prevent a malfunction.

(4) Single transfer performed with two to three channels and burst transfer

Perform (a) to prevent a malfunction. For the channel used for burst transfer, set the same offset to DMSARn and DMDARn, or set 1 to DMCCRn.USEXFSZ. (Set the on-chip FIFO to be shared with multiple DMA channels by which no data remains in FIFO.)

In dual address burst transfer, when 0 is set to the transfer size mode bit and the offset value is different between source address and destination address, data may remain in FIFO. (See 8.3.8.2 Burst Transfer During Dual address Transfer.) When a bus error occurs during single transfer for which workaround (a) is performed, data is erased if it remains in FIFO during burst transfer. This occurs because FIFO is reset due to a bus error. Since DMAC does not detect data erasing, you need to set FIFO to contain no data.

24.4 Note on PCI Controller

- Set four target spaces, that are MEM0, MEM1, MEM2 and IO contained in PCI controller, so that address windows are not duplicated. See 10.3.5 Target Access.

- We recommend to set 0 to G2PTOCNT. See 10.4.14 G2P Timeout Count Register.

- When TX4938 is the PCI target, access in dual address cycle is disabled. See 10.3.3 Supported PCI Bus Commands.

- Notes on Register Read by PCI Controller

  Read the Power Status (PS) field in the PCISTATUS register (10.4.17) by the following procedures.

  (1) Normal method

  After checking that P2GSTATUS.PMCS bit is set, read the PCISTATUS.PS field.

  (2) Method to read at any timing

  To read the PCISTATUS.PS field directly without using the method (1), read the field twice consecutively. The same value which is read twice consecutively is applied.

- When writing to PCI bus by PDMAC of PCIC, don't read on-chip SRAM or register in the controller connected to G-Bus.

- Restrictions on use of the broken master function in the PCI controller

[Restrictions]

When the broken master function in the PCI controller is used, the master which is not broken may be incorrectly acknowledged to be broken.

- Don't use the broken master function

or

- To use the broken master function, use only the high level containing Master A, B, C or D.

[Overview]

When the broken master function in the PCI controller is used, the master which is not broken may be incorrectly acknowledged as the broken master.

The broken master detection function of the on-chip PCI bus arbiter detects the PCI master, which does not start access though it has the bus, as the broken master and removes that PCI master from the bus arbitration sequence.

When the broken master detection function detects the broken master, it may also detect the non-broken master and acknowledge it as a broken master.

<Conditions>

This problem occurs when the following conditions are satisfied.

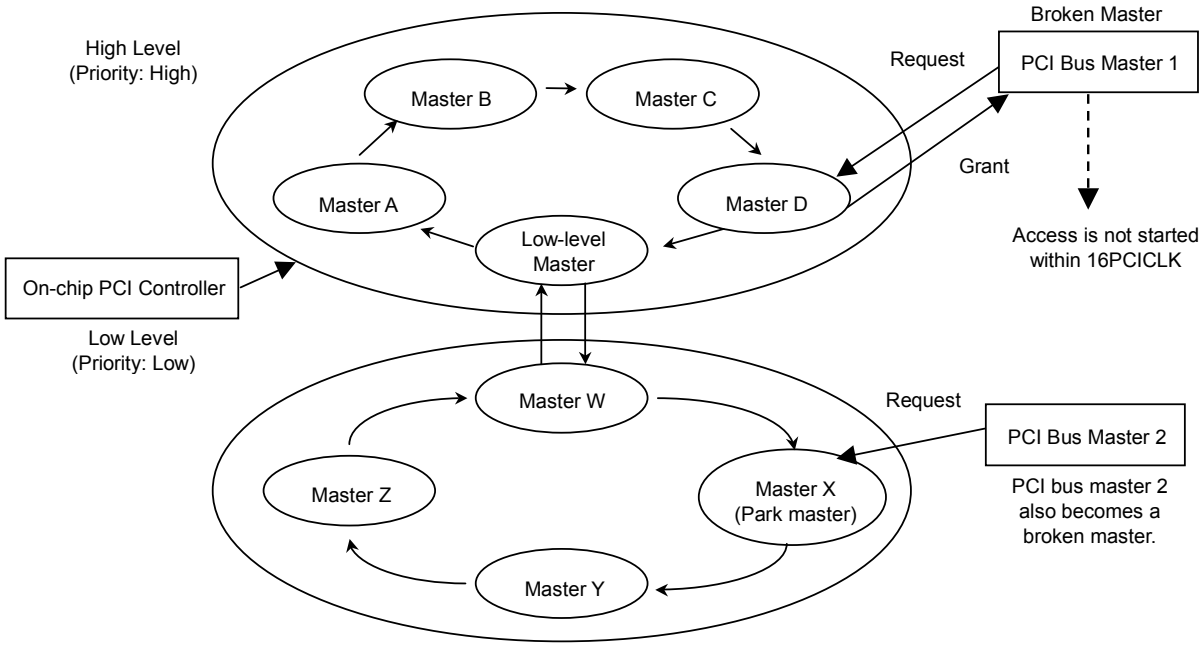
(1) The broken master detection function is enabled (BMCEN=1).

The broken master detection function is enabled or disabled with the BMCEN bit of the PCI bus arbiter configuration register (PBACFG). The default value is 0 which disables the function.

(2) The bus masters are assigned to the high level and low level in the on-chip PCI bus arbiter.

The bus master is assigned to the bus arbiter with the PCI Bus Arbiter Request Port Register (PBAREQPORT).

(3) When the PCI bus master connected to the high level is detected as the broken master, the master connected to the low level requests the bus mastership, which is the highest-priority request in the low level. (That is Master W immediately after a reset or the master which acquires the bus mastership most recently in the low level when the fixed park master mode is not set, or Master W that is the park master in the fixed park master mode.)



PCI Arbitration Priority

[Workarounds]

There are two workarounds for this problem.

(1) Don't use the broken master function.

Don't set the BMCEN bit of the PBACFG register.

(2) When using the broken master function, use only the high level containing Master A, B, C and D.

- Note on use of the PCI boot function

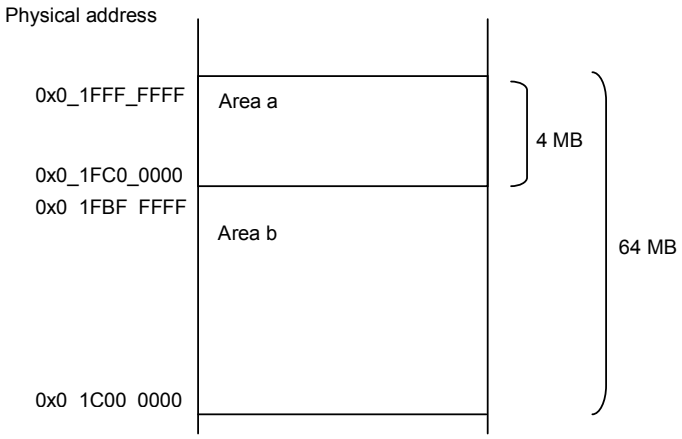
[Restriction]

Don't perform PCI boot in default setting.

[Violation]

When performing PCI boot in the default setting, 64 Mbyte memory space that is 0x0_1C00_0000 to 0x0_1FFF_FFFF (physical address, the area a and b in the figure shown below) is assigned to the PCI bus, and not 4 Mbyte memory space that is 0x0_1FC0_0000 to 0x0_1FFF_FFFF (physical address, the area a).

PCI boot is performed correctly. When the memory space that is 0x0_1C00_0000 to 0x0_1FBF_FFFF (physical address, the area b) is accessed after boot, malfunction occurs due to conflict between PCIC and the other controller such as SDRAMC which assigns this memory space. There is no problem when the memory space that is 0x0_1C00_0000 to 0x0_1FBF_FFFF (physical address, the area b) is not accessed after boot.



<Conditions>

This malfunciton occurs when the memory space that is 0x0_1C00_0000 to 0x0_1FBF_FFFF (physical address) is accessed after boot. When there is no memory space that is 0x0_1C00_0000 to 0x0_1FBF_FFFF (physical address) other than PCIC, this malfunction does not occur.

[Workaround]

To assign correct memory space, change the value of G2P Memory Space 2 Address Mask Register (G2PPM2MASK) to the following value.

	Default value		Change to
G2PM2MASK(0xD148)	0x003f_fff0	->	0x0003_fff0

- Restriction when Initiator Write by PDMAC and Target Read conflict.

[Restriction]

Don't perform Target Read from a register on the G-Bus when the condition is the following <Conditions>.

[Violation]

When an Initiator Write transaction using PDMAC (PCI Dedicated DMA Controller) mounted in the PCI Controller of the target product and a Target Read transaction to the target product by the a device on the PCI Bus conflict, there are cases when the Target Read data is corrupted.

<Conditions>

(1) In the PCI Controller of the above target product:

A. PDMAC performs an Initiator Write transaction to a device on the PCI Bus.

B. Device on the PCI Bus becomes the Bus Master and performs a Target Read on the target product.

When the two above accesses conflict,

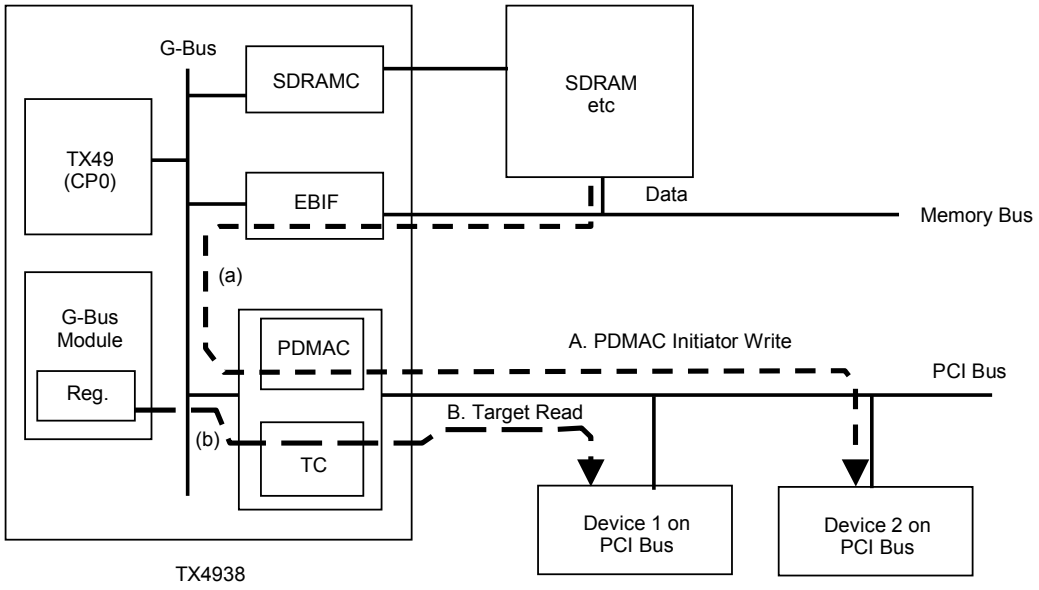
(2) The internal bus (G-Bus) of the target product is accessed continuously in the following order.

(a) PDMAC reads the Initiator Write data on the G-Bus.

(b) TC (Target Controller) reads data from the G-Bus because of a Target Read request.

(3) The target of the Target Read in (2) is a register on the G-Bus

However, there is no corresponding register on the Internal Bus (IM-Bus).



[Workarounds]

Do not perform Target Read access from a register on the G-bus or Internal SRAM area under the above conditions. The register on the G-Bus is a register with the 0x5000 to 0x5FFF, 0x6000 to 0x67FF and 0x7000 to 0xEFFF.

24.5 Notes on Serial I/O Port

- Restrictions on use of the break function in SIO

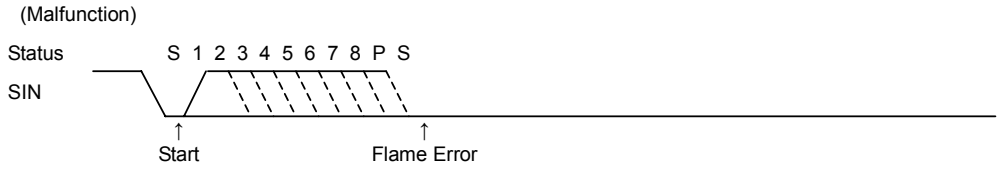
[Restriction]

To transmit breaks to TX4938, synchronize breaks to the start bit. Set consecutively the transmit data to Low immediately after the start bit.

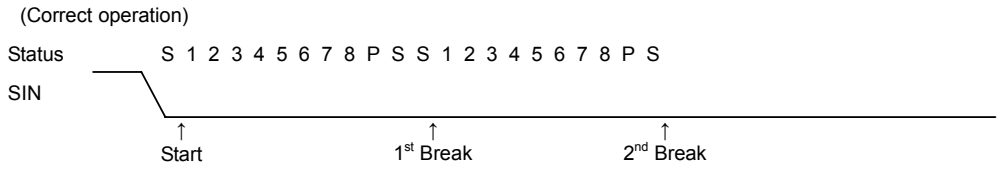
[Violation]

When the transmitting end transmits breaks to TX4938 in the middle of a transmission, TX4938 detects the first flame error only, but not breaks.

When the break reception is synchronized to the start bit (the receive data is consecutively set to Low immediately after the start bit), breaks can be detected correctly.



* Since the start bit is not recognized after a flame error, the receiving status is stopped in idle state.



* When the receiving data is consecutively set to Low immediately after the start bit is recognized, breaks can be detected.

When breaks are transmitted to TX4938, it may not receive breaks.

This malfunction may occur when the transmitting end transmits breaks to TX4938 in the middle of a transmission.

[Workarounds]

To transmit breaks to TX4938, synchronize breaks to the start bit (set the transmitting data consecutively to Low immediately after the start bit).

24.6 Notes on Ether Controller

- Initializing SRAM containing Ether controller

Note: If you use SRAM without initialization, parity errors may be detected incorrectly during packet receiving. To prevent this malfunction, perform the following initialization.

(1) Preparation

Set TestMode (bit13) of the DMA_Ctl (0x00) register. (Write 1)

(2) Initialization of on-chip memory

Perform the following for all addresses (0x000 to 0x3ff) of on-chip SRAM.

(i) Set the address of on-chip SRAM to ARC_Adr (0x60).
(ARC_Loc field: 0x000 to 3ff)

(ii) Write 0x00000000 to ARC_Data (0x64) (32-bit write)

You can check the data which was written.

Read ARC_Data (0x64) (32-bit read)

(3) Release test mode

Reset TestMode (bit 13) of the DMA_Ctl (0x00) register. (Write 0)

25. Parts Number when Ordering

	Parts Number	Package	Maximum Operating Frequency
TX4938	TMPR4938XBG-300	484-pin PBGA	300 MHz
	TMPR4938XBG-333	484-pin PBGA	333 MHz

Appendix A TX49/H3 Core Supplement

This section explains items that are unique to the TX4938 of the TX49/H3 Core. Please refer to the "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture" for more information regarding the TX49/H3 Core.

A.1 Processor ID

PRId Register values of the TX4938 TX49/H3 Core are as follows.

Processor Revision Identifier Register: 0x0000 2D30

FPU Implementation/Revision Register (FCR0): 0x0000 2D30

These values may be changed at a later date. Please contact the Toshiba Engineering Department for the most recent information.

A.2 Interrupts

Interrupt signalling of the on-chip interrupt controller is reflected in bit IP[2] of the Cause Register in the TX49/H3 Core. In addition, interrupt causes are reflected in other bits of the IP field. Please refer to Section "15.3.5 Interrupt notification" for more information.

A.3 Bus Snoop

The Bus Snoop function is not used with the TX4938 due to restrictions of the Bus Snoop specification.

A.4 Halt/Doze mode

The Doze mode is not necessary when the Bus Snoop function is not used. Please use the Halt mode, which further reduces power consumption. Clearing the HALT bit of the Config Register makes it possible to shift to the Halt mode by executing the WAIT instruction.

A.5 Memory access order

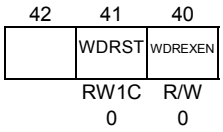
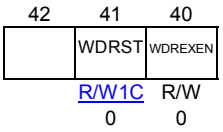
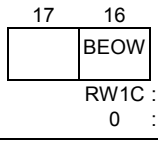
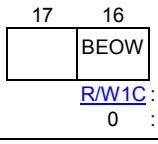
The TX49/H3 Core has a 4-stage Write buffer, the PCI Bus Bridge (PCI Controller) has 4 stages for initiator access, and has a 2-stage Post Write buffer (Write buffer) for target access.

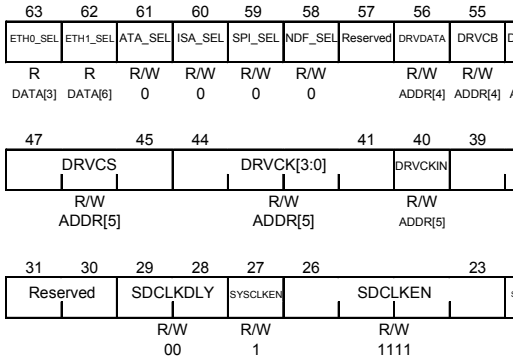
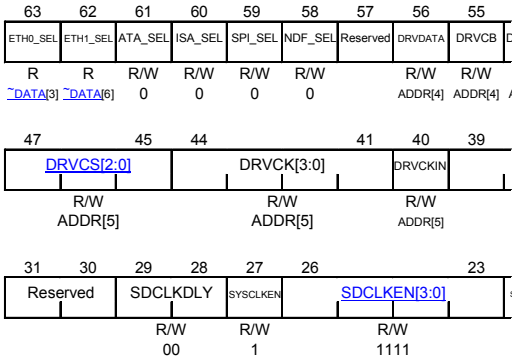
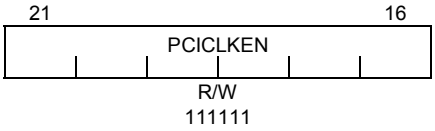
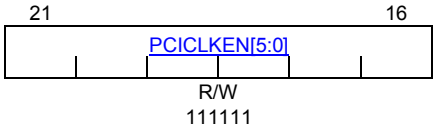
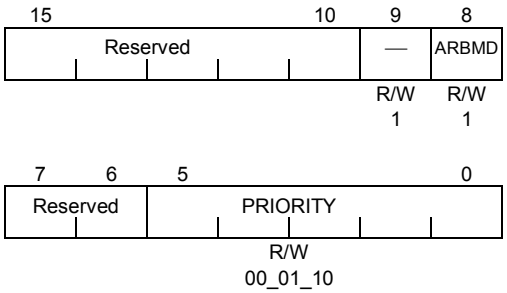
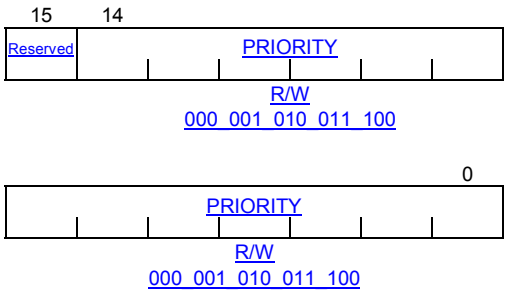
When data enters the Write buffer of the TX49/H3 Core, Cache Refill Read operations that do not match the address of that data after the Write is issued may be issued to the internal bus (G-Bus) before the Write. Other accesses are issued in order.

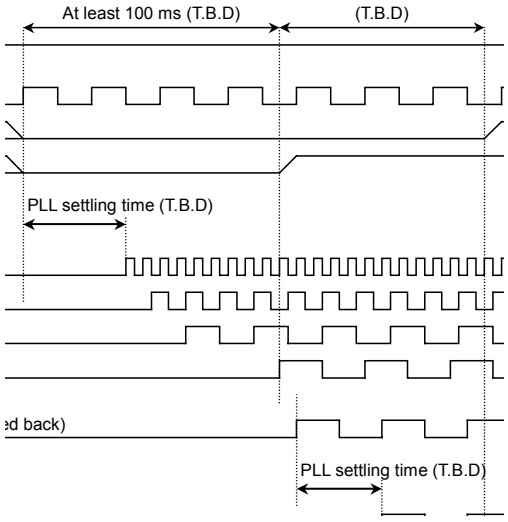
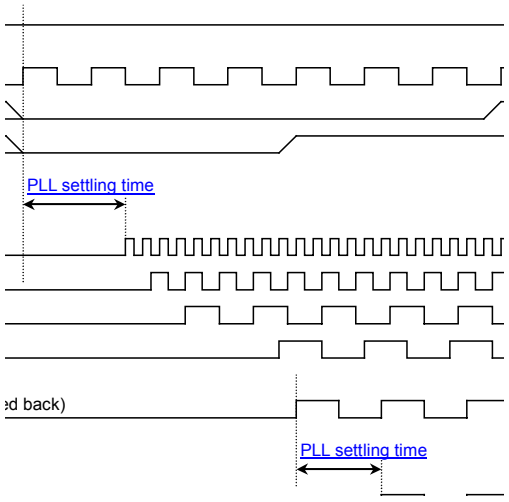
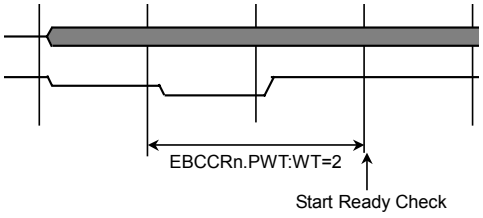
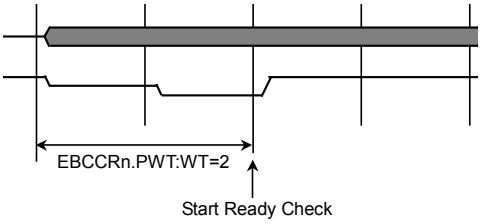
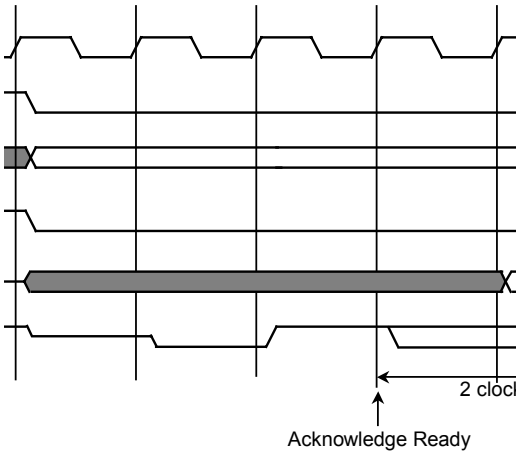
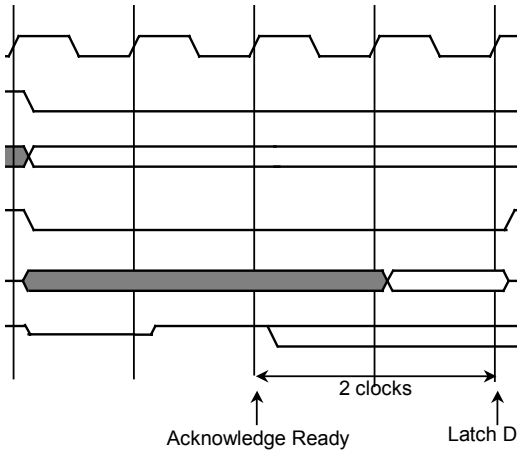
Executing the SYNC instruction guarantees that bus access invoked by a load/store instruction previously executed will be complete on the internal bus.

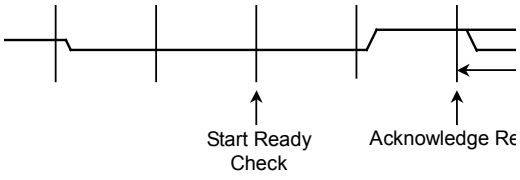
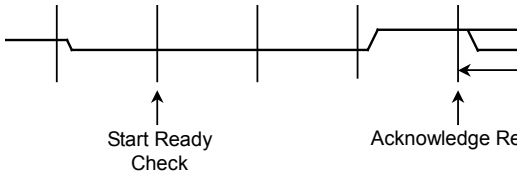
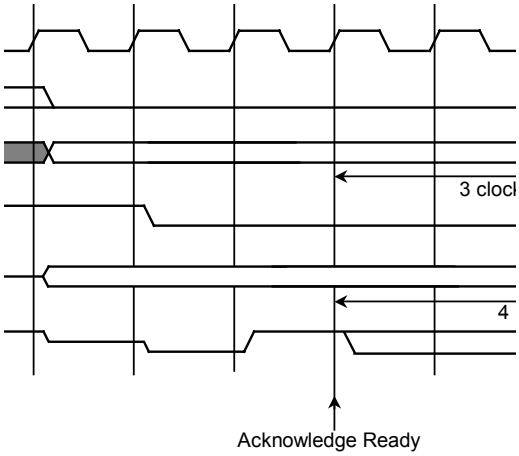
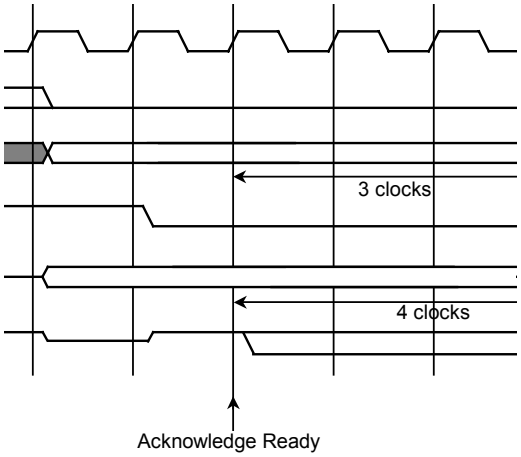
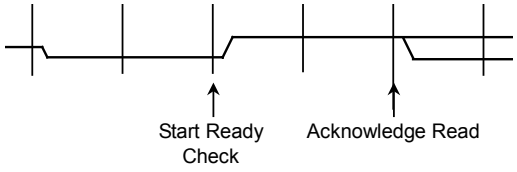
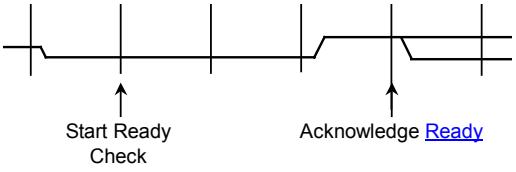
The PCI Bus Bridge is issued by the issue destination bus in the order all bus accesses are issued on the issue source bus. Please refer to "10.3.6 Post Write Function" for more information regarding methods for guaranteeing the completion of Write transactions of the Post Write Buffer.

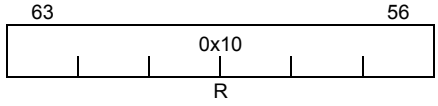
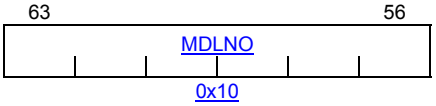
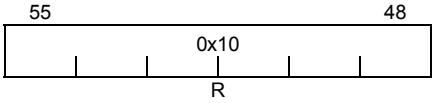
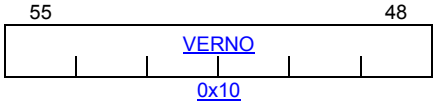
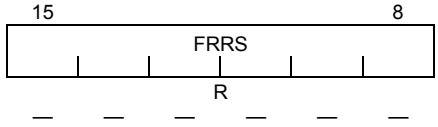
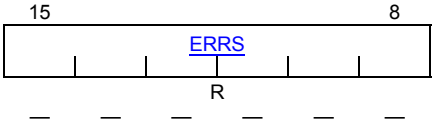
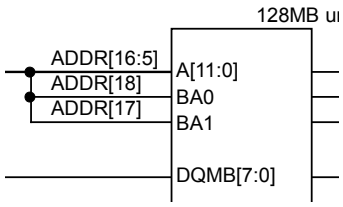
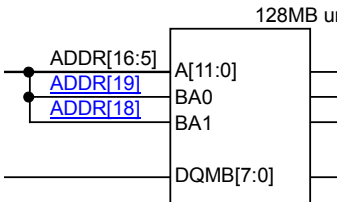
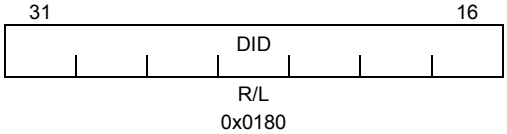
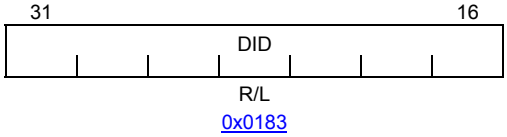
TMPR4938 Revision History

Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
2	References 64-bit TX System RISC TX49/H3 Core Architecture User's Manual	 64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture
1-1	Modified line 5 of the body text in Section 1.1, *Overview* For details of the TX49/H3 core such as instruction sets, see "64-bit TX System RISC TX49/H3 Core Architecture".	 For details of the TX49/H3 core such as instruction sets, see "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture".
3-14	Table 3.2.2 Boot Configuration Specified with the ADDR[19:0] Signals (1/2) Modified the description of the ADDR[17] and ADDR[15]. Reserved	 Reserved Used for testing. This signal will not be set to 0 upon booting.
4-1	Modified line 5 of the introduction in Chapter 4, *Address Mapping* Please refer to "64 bit TX System RISC TX49/H3 Core Architecture" about the details of mapping to a physical address from the virtual address of TX49/H3 core.	 Please refer to "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture" about the details of mapping to a physical address from the virtual address of TX49/H3 core.
5-3	Figure 5.2.1 Chip Configuration Register (1/3) 	 
5-3	Figure 5.2.1 Chip Configuration Register (1/3) 	 
5-3	Figure 5.2.1 Chip Configuration Register (1/3) Read/write attribute of the WDRST (Watchdog Reset Status) bit RW1C	 R/W1C
5-4	Figure 5.2.1 Chip Configuration Register (2/3) Read/write attribute of theBEOW (Write-Access Bus Error) bit RW1C	 R/W1C

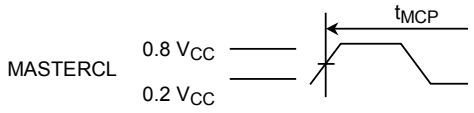
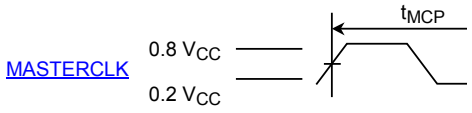
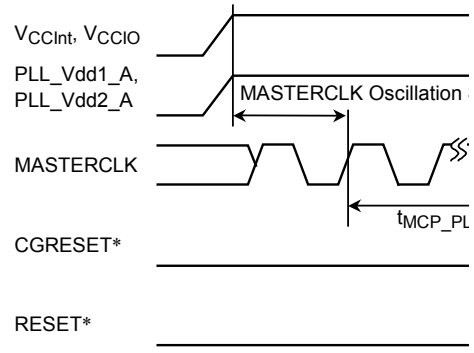
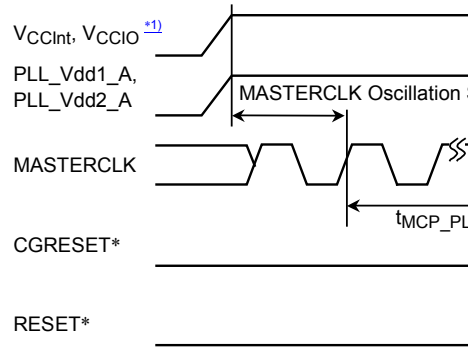
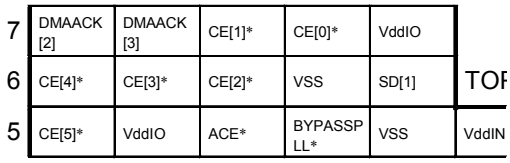
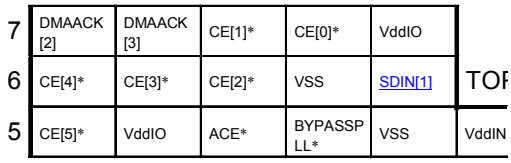
Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
5-7	Figure 5.2.3 Pin Configuration Register (1/3) 	
5-7	Figure 5.2.3 Pin Configuration Register (1/3) 	
5-7	Figure 5.2.3 Pin Configuration Register (1/3) Initial value of the ETH0_SEL (Status of shared pin) bit DATA[3]	 ~DATA[3]
5-7	Figure 5.2.3 Pin Configuration Register (1/3) Initial value of the ETH1_SEL (Status of shared pin) bit DATA[6]	 ~DATA[6]
5-8		Figure 5.2.3 Pin Configuration Register (2/3) Added a note to the description of the DRVCB (CB Signal Control) bit Note: CB[7:0]* share pins with PIO[15:8], E0TXD[3:0], E0RXD[3:0]. The driving capability of these pins are below. CB[7:0], E0TXD[3:0], E0RXD[3:0]: 8 mA or 16 mA PIO[15:8]: 8 mA only
5-14	Figure 5.2.6 G-Bus Arbiter Control Register 	

Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
6-6	Figure 6.3.1 Power-On Sequence 	
7-11		Added the following text to line 8 to Section 7.3.6.3, *Ready Mode* When the number of wait cycles is 0, READY check is started in 1 cycle after asserting the CE* signal. When the number of wait cycles is other than zero, after waiting only for the specified number of cycles, READY check is started.
7-11	Figure 7.3.3 Ready Mode 	
7-18	Figure 7.3.13 Ready Input Timing (Read Cycle) 	

Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
7-18	Figure 7.3.13 Ready Input Timing (Read Cycle) 	
7-19	Figure 7.3.14 Ready Input Timing (Write Cycle) 	
7-19	Figure 7.3.14 Ready Input Timing (Write Cycle) 	
7-20	Modified line 2 of Section 7.3.9, *ISA /ATA Mode* Since the pins used in ISA/ATA mode are multiplexed pins, select ISA/ATA before use of these pins. (Refer to "5.2.3 Pin Configuration Register".)	Since the pins used in ISA/ATA mode are multiplexed pins, select ISA/ATA before use of these pins. Since the signal of CE* is not used in this mode, when ETHER1 is chosen, this mode can be used by channels 5, 6 and 7. (Refer to "5.2.3 Pin Configuration Register".)
7-20	Modified line 3 of the body text in Section 7.3.9.1, *Address space* When access to the ISA I/O space is performed, IOR* and IOW* control signals are valid, and OE* and SWE* are invalid.	When access to the ISA I/O space is performed, IOR* and IOW* control signals are valid, and OE*, SWE* and CE* are invalid.
7-31	Figure 7.5.3 Double-word Single Write (PWT: WT=1, SHWT=0, Normal, 32-bit Bus)	Figure 7.5.3 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)
8-13	Table 8.3.3 Channel Register Setting Restrictions During Dual Address Transfer <DMSAIRn> 8/0/-8	8/0/-8 †

Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
8-13		Table 8.3.3 Channel Register Setting Restrictions During Dual Address Transfer Added the following note. †: When DMSAIRn is set to 0, read access from source device is performed only one time per transmission specified by DMCCRn.XFSZ. For this reason, transfer can not be performed burst transfer to the I/O device which performs FIFO operation.
9-23	Figure 9.4.4 ECC Control Register (1/2) 	
9-23	Figure 9.4.4 ECC Control Register (1/2) 	
9-23	Figure 9.4.4 ECC Control Register (1/2) Modified the description of the DEEC (Diagnostic ECC) field. The value set by this field is output from CB[7:0] as the check code when the ECCDM bit is set to “Enable.”	The value set by this field is output from CB[7:0] as the check code when the DM bit is set to “Enable.”
9-25	Figure 9.4.5 ECC Status Register 	
9-42	Figure 9.6.2 168-pin DIMM Connection Example 	
10-26	Figure 10.4.1 ID Registers 	
10-26	Figure 10.4.1 ID Registers Modified the description of the DID (Device ID) field. Device ID (Default: 0x0180)	Device ID (Default: 0x0183)

Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
11-22	Figure 11.4.7 Baud Rate Control Register Modified the description of the BCLK (Baud Rate Generator Clock) field. 00: Select prescalar output T0 (IMBUSCLK/2) 01: Select prescalar output T2 (IMBUSCLK/8) 10: Select prescalar output T4 (IMBUSCLK/32) 11: Select prescalar output T6 (IMBUSCLK/128)	00: Select prescalar output T0 (fc/2) 01: Select prescalar output T2 (fc/8) 10: Select prescalar output T4 (fc/32) 11: Select prescalar output T6 (fc/128)
14-11	Table 14.3.7 Mic DMA Buffer Format in Big-endian Mode Address offset | +0 | +1 z | #0H | #0L +4 | #1H | #1L +8 | #2H | #2L : | : | :	Address offset | +0 | +1 +0 | #0H | #0L +4 | #1H | #1L +8 | #2H | #2L : | : | :
16-26	Deleted the following text from Section16.3.8.5, *Frame reception procedure* Usually, before setting the above code, the DMA Controller is initialized by storing a valid address in the Buffer List Form Pointer Register or initializing a Free Descriptor Area Register and a Free Descriptor Size Register for example.	Usually, before programming the above registers, the DMA Controller is initialized.
16-33	Modified line 20 of Section16.3.9.2, *Initial setup of DMA and MAC* • DMA Transmission Control Register: To customize the Burst sizeC	• DMA Control Register: To customize the Burst size
16-34	Modified line 9 of Section, 16.3.9.3.2, *Initializing the buffer list* You can use the Reception Buffer Fragment Size Register to globally pack a buffer.	You can use the Reception Fragment Size Register to globally pack a buffer.
16-35	Modified line 6 of Section16.3.9.5, *Receiving frames* • Initialize the Reception Frame Pointer Register to the address of the dummy frame descriptor in the free descriptor area.	• Initialize Free Descriptor Area Base Register to the address of the dummy frame descriptor in the free descriptor area.
16-36	Modified line 4 of Section, 16.3.9.5.2, *Releasing buffers* The Control field of the Buffer Fragment Size Register or Frame Descriptor controls the method of mapping the buffer.	The Control field of the Reception Fragment Size Register or Frame Descriptor controls the method of mapping the buffer.
21-2	Modified line 2 of Section 21.2.1, *JTAG Controller and Register* Please refer to the TX49/H3 Core Architecture Manual for all other portion not covered here.	Please refer to the "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture" for all other portion not covered here.
21-3	Modified line 3 of 21.2.2, *Instruction Register* Refer to the TX49/H3 Core Architecture Manual for more information regarding each instruction.	Refer to the "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture" for more information regarding each instruction.

Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
21-3	Table 21.2.1 Bit Configuration of JTAG Instruction Register Refer to the TX49/H3 Core Architecture Manual	 Refer to the "64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture"
22-5	Figure 22.5.1 Timing Diagram: MASTERCLK 	
22-5	Figure 22.5.2 Timing Diagram: Power On Reset  Note 1: VCCInt and VCCIO must start up simaltaneously, or VCCInt must be first. Note 2: The difference of the stand up time of a power supply within in 100 m seconds.	 *1) VCCInt and VCCIO must start up simultaneously, or VCCInt must be first. The difference of the stand up time of a power supply within in 100 m seconds.
23-2	Figure 23.1.1 Pinout Diagram (1/2) 	
23-4	Table 23.1.1 Pin Cross Reference by Pin Number (1/2) Modified the pin name of the E6 pin. SD[1]	 SDIN[1]
23-7	Table 23.1.2 Pin Cross Reference by Pin Name (2/2) Modified the pin name of the E6 pin. SD[1]	 SDIN[1]
24-1	24. Notes on Use of TMPR4938 24.1 Notes on External Bus Controller 24.2 Notes on DMA Controller 24.3 Note on PCI Controller 24.4 Notes on Serial I/O Port 24.5 Notes on Ether Controller	 24.1 Notes on TX49/H3 Core 24.2 Notes on External Bus Controller 24.3 Notes on DMA Controller 24.4 Note on PCI Controller 24.5 Notes on Serial I/O Port 24.6 Notes on Ether Controller
24-9		24.4 Note on PCI Controller The section "Restriction when Initiator Write by PDMAC and Target Read conflict" is added.

Page	Rev 1.2 Manual	Changes and Additions to Rev 1.2
A-1	Modified line 2 of the introduction of Appendix A, *TX49/H3 Core Supplement* Please refer to the “64-bit TX System RISC TX49/H3 Core Architecture User’s Manual” for more information regarding the TX49/H3 Core.	Please refer to the “64-bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture” for more information regarding the TX49/H3 Core.